Analysing Illumina bead-based data using beadarray

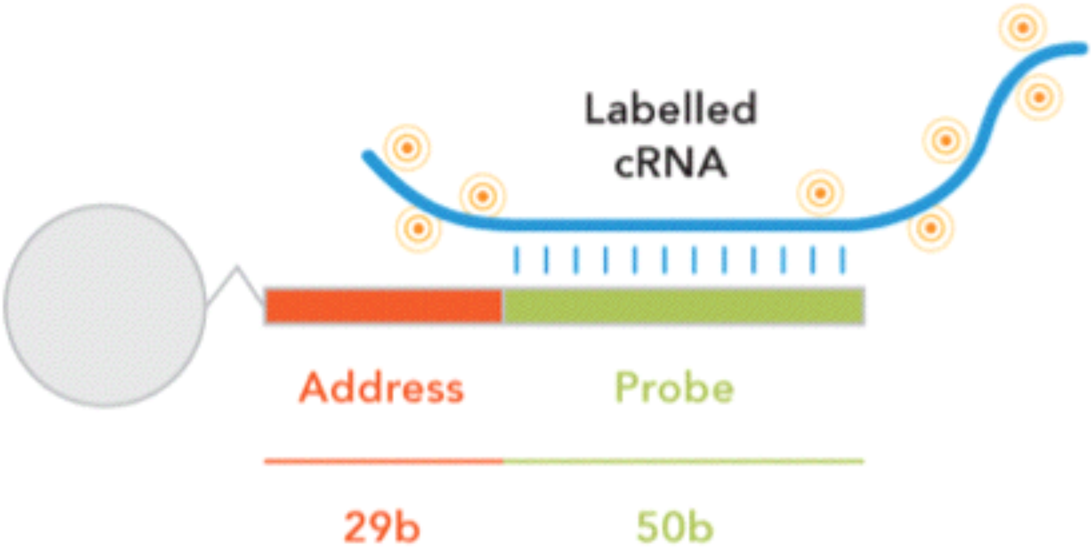
Mark Dunning

6th August 2007

UNIVERSITY OF CAMBRIDGE

CANCER RESEARCH UK

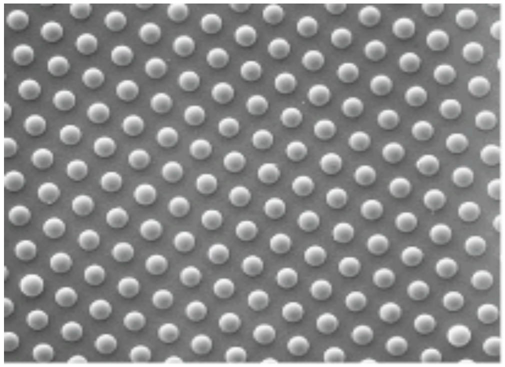
The Bead

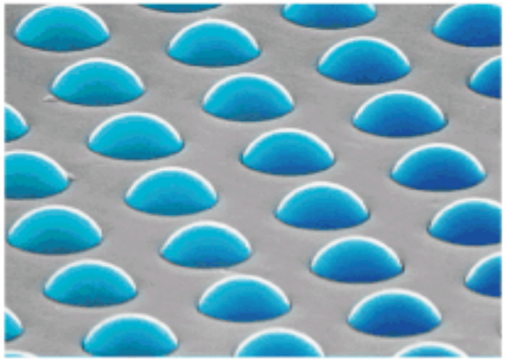


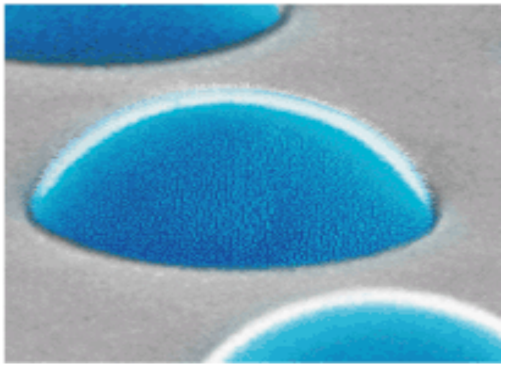
Each silica bead is 3 microns in diameter

700,000 copies of same probe sequence are covalently attached to each bead for hybridisation & decoding

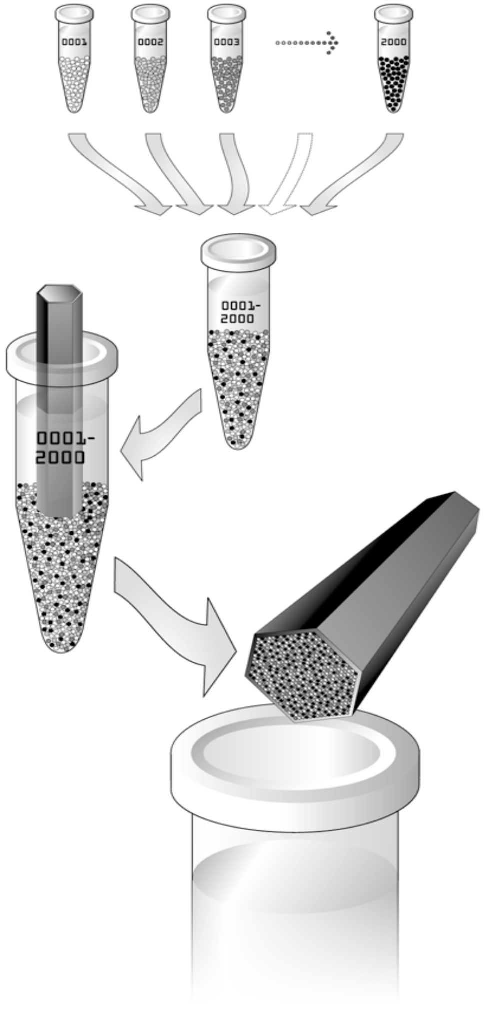
Beads in Wells





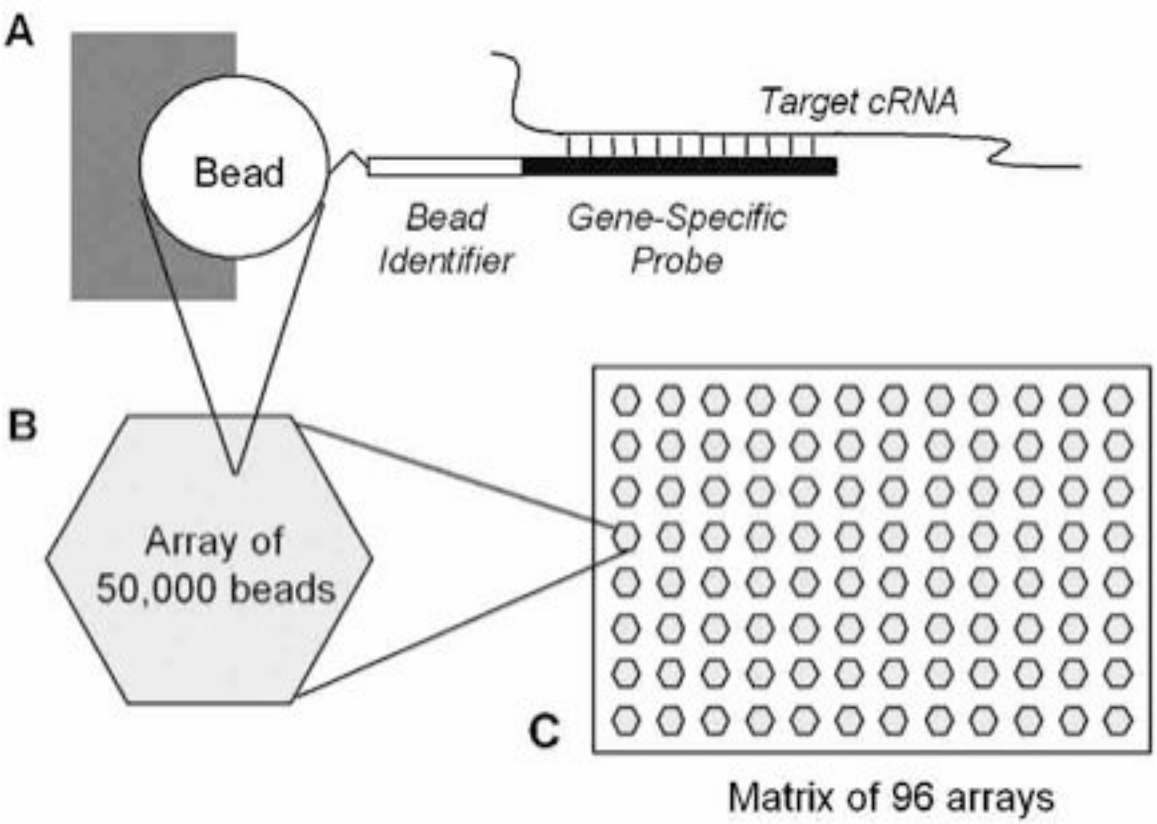


Bead Preparation and Array Production



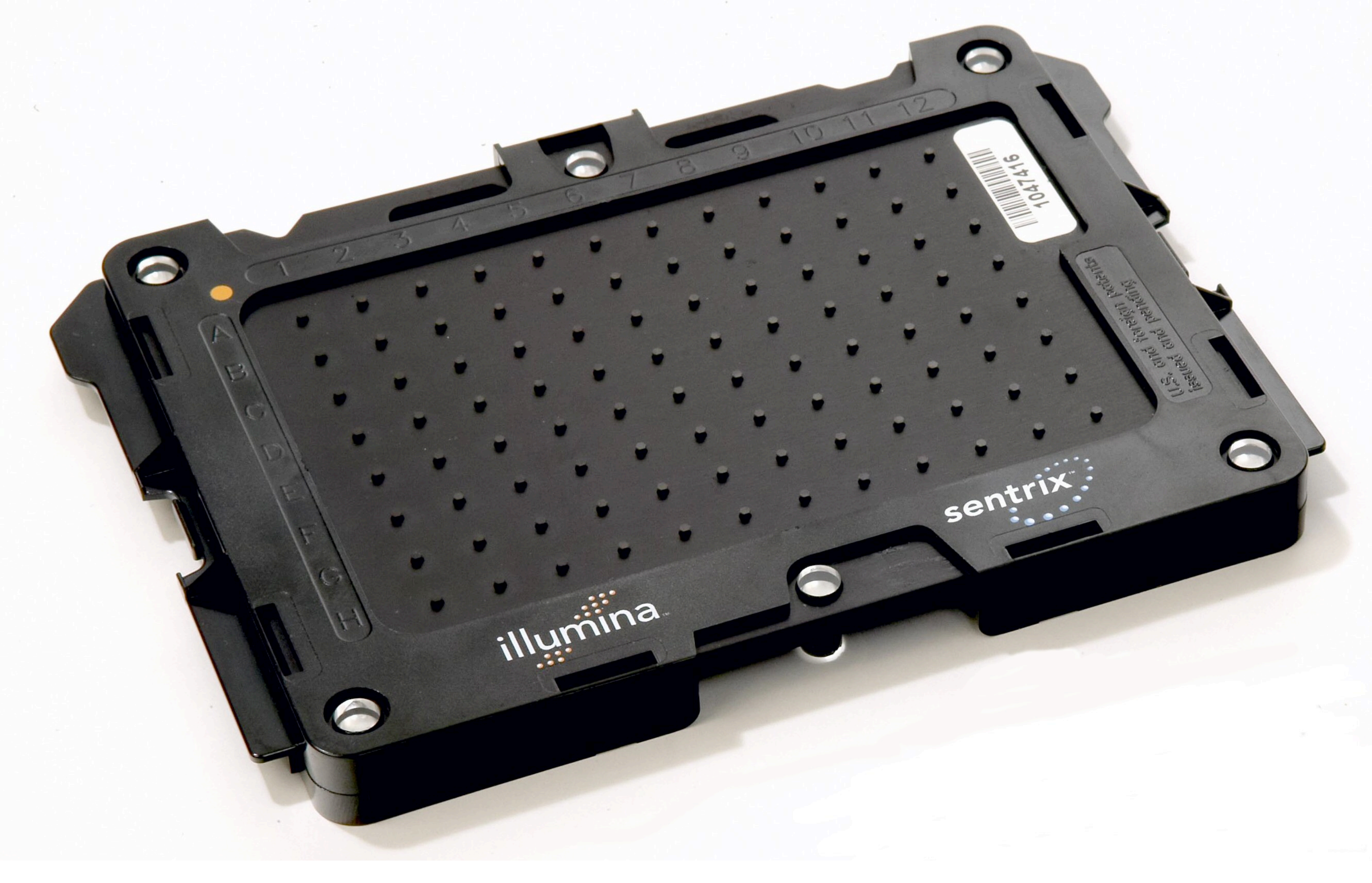
- Bead pools produced containing 384 to 24,000 bead types
- Wells created in either fibre-optic bundle (hexagon) or chip (rectangle) & exposed to array
- Beads self-assemble into wells to form **randomly arranged** array of beads
- *Average of 30 beads of each type*
- Each array produced separately

Combining Arrays - The SAM

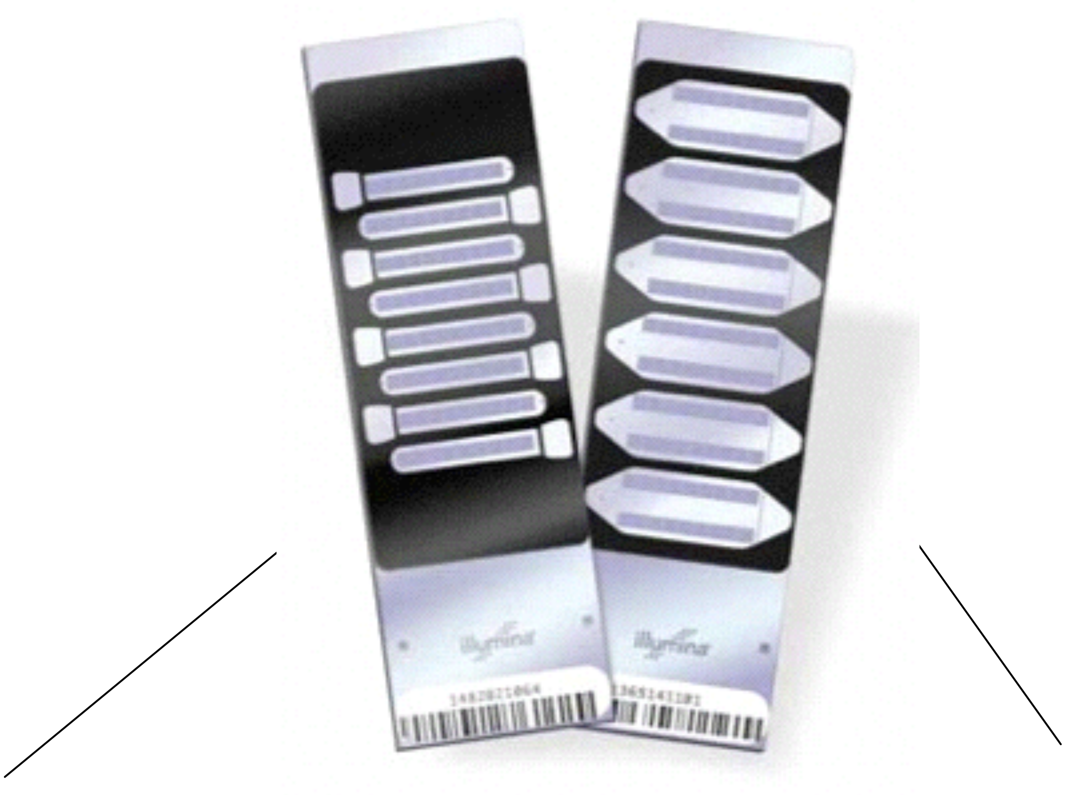


~1500 bead types on array ~30 of each type
1 array = 1 sample or treatment
96 arrays processed in parallel - **High throughput**

The SAM



Combining Arrays - BeadChips



RefSeq BeadChip

8 arrays per chip 1 strip = 1 array

24,000 bead types from RefSeq database x 30 reps on each array

Whole Genome

6 arrays per chip: 2 strips = 1 array

48,000 bead types (24,000 RefSeq + 24,000 supplemental) on each array

Raw Data

Illumina’s scanning software (BeadScan) produces encrypted files (.idat, .locs etc) which are read by their proprietary analysis software (BeadStudio)

However, with modifications BeadScan you can also get more useful (readable) files for each array on a SAM or *strip* on a BeadChip

-Text file giving the identity and location of each individual bead - with 50,000 rows for SAM ~ 1.1 million for BeadChip

-TIFF images (and not jpegs)

We refer to the TIFF and text files as the **bead level data** for an array

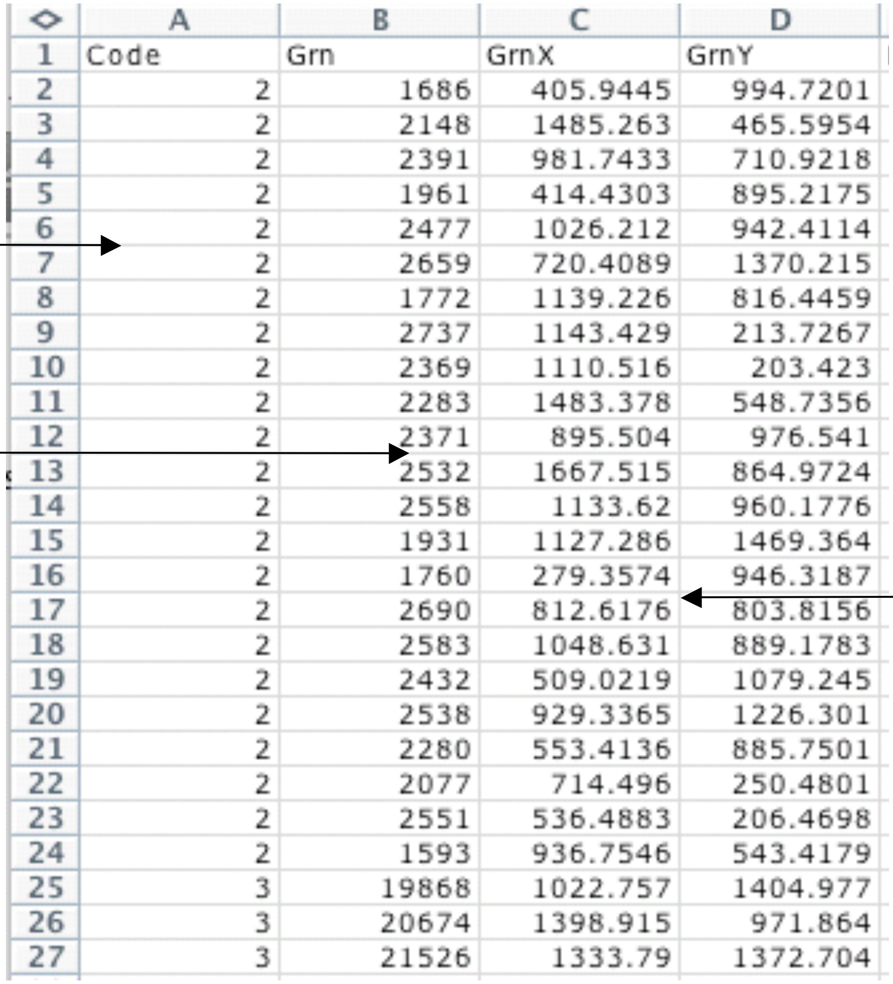
Bead Level Text Files

Example of a bead level text file

ProbeID

Corrected Intensity

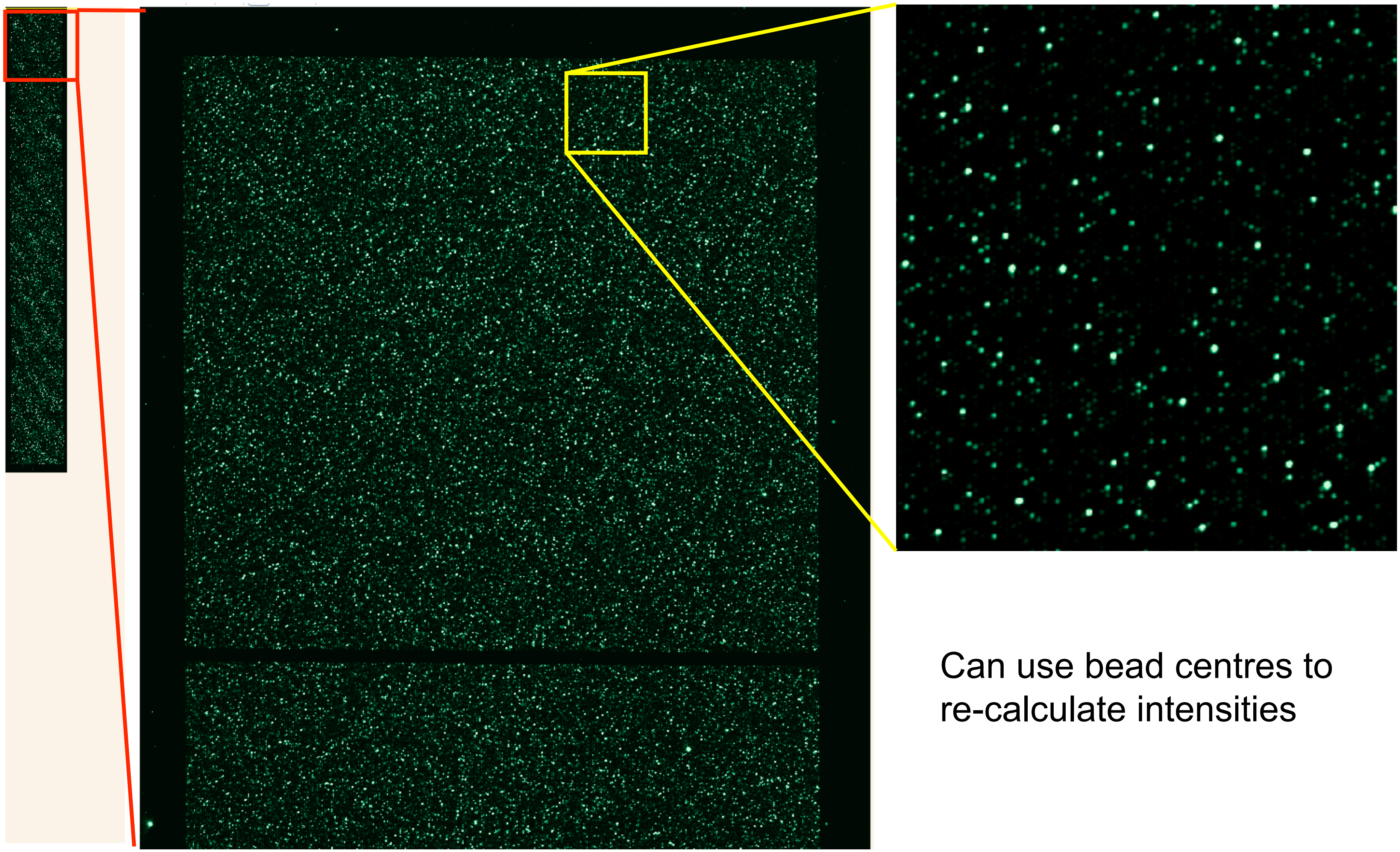
Bead Centre



◇	A	B	C	D
1	Code	Grn	GrnX	GrnY
2	2	1686	405.9445	994.7201
3	2	2148	1485.263	465.5954
4	2	2391	981.7433	710.9218
5	2	1961	414.4303	895.2175
6	2	2477	1026.212	942.4114
7	2	2659	720.4089	1370.215
8	2	1772	1139.226	816.4459
9	2	2737	1143.429	213.7267
10	2	2369	1110.516	203.423
11	2	2283	1483.378	548.7356
12	2	2371	895.504	976.541
13	2	2532	1667.515	864.9724
14	2	2558	1133.62	960.1776
15	2	1931	1127.286	1469.364
16	2	1760	279.3574	946.3187
17	2	2690	812.6176	803.8156
18	2	2583	1048.631	889.1783
19	2	2432	509.0219	1079.245
20	2	2538	929.3365	1226.301
21	2	2280	553.4136	885.7501
22	2	2077	714.496	250.4801
23	2	2551	536.4883	206.4698
24	2	1593	936.7546	543.4179
25	3	19868	1022.757	1404.977
26	3	20674	1398.915	971.864
27	3	21526	1333.79	1372.704

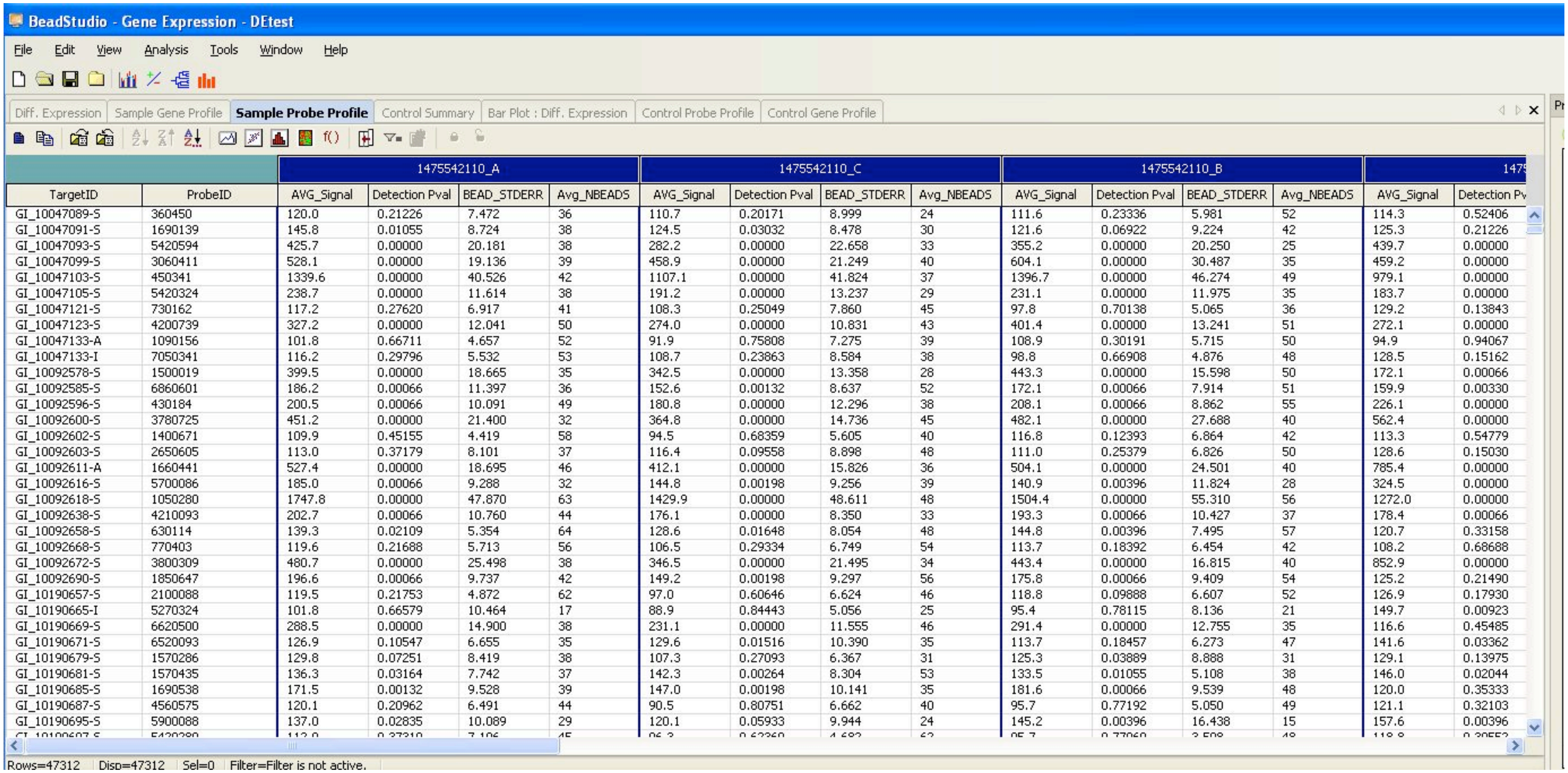
Information for **“all”** beads on an array (50,000 or 1 million rows)
Sometimes outliers or non-decoded beads are removed

TIFF images



Can use bead centres to re-calculate intensities

BeadStudio output



		1475542110_A				1475542110_C				1475542110_B				1475	
TargetID	ProbeID	AVG_Signal	Detection Pval	BEAD_STDERR	Avg_NBEADS	AVG_Signal	Detection Pval	BEAD_STDERR	Avg_NBEADS	AVG_Signal	Detection Pval	BEAD_STDERR	Avg_NBEADS	AVG_Signal	Detection Pv
GI_10047089-S	360450	120.0	0.21226	7.472	36	110.7	0.20171	8.999	24	111.6	0.23336	5.981	52	114.3	0.52406
GI_10047091-S	1690139	145.8	0.01055	8.724	38	124.5	0.03032	8.478	30	121.6	0.06922	9.224	42	125.3	0.21226
GI_10047093-S	5420594	425.7	0.00000	20.181	38	282.2	0.00000	22.658	33	355.2	0.00000	20.250	25	439.7	0.00000
GI_10047099-S	3060411	528.1	0.00000	19.136	39	458.9	0.00000	21.249	40	604.1	0.00000	30.487	35	459.2	0.00000
GI_10047103-S	450341	1339.6	0.00000	40.526	42	1107.1	0.00000	41.824	37	1396.7	0.00000	46.274	49	979.1	0.00000
GI_10047105-S	5420324	238.7	0.00000	11.614	38	191.2	0.00000	13.237	29	231.1	0.00000	11.975	35	183.7	0.00000
GI_10047121-S	730162	117.2	0.27620	6.917	41	108.3	0.25049	7.860	45	97.8	0.70138	5.065	36	129.2	0.13843
GI_10047123-S	4200739	327.2	0.00000	12.041	50	274.0	0.00000	10.831	43	401.4	0.00000	13.241	51	272.1	0.00000
GI_10047133-A	1090156	101.8	0.66711	4.657	52	91.9	0.75808	7.275	39	108.9	0.30191	5.715	50	94.9	0.94067
GI_10047133-I	7050341	116.2	0.29796	5.532	53	108.7	0.23863	8.584	38	98.8	0.66908	4.876	48	128.5	0.15162
GI_10092578-S	1500019	399.5	0.00000	18.665	35	342.5	0.00000	13.358	28	443.3	0.00000	15.598	50	172.1	0.00066
GI_10092585-S	6860601	186.2	0.00066	11.397	36	152.6	0.00132	8.637	52	172.1	0.00066	7.914	51	159.9	0.00330
GI_10092596-S	430184	200.5	0.00066	10.091	49	180.8	0.00000	12.296	38	208.1	0.00066	8.862	55	226.1	0.00000
GI_10092600-S	3780725	451.2	0.00000	21.400	32	364.8	0.00000	14.736	45	482.1	0.00000	27.688	40	562.4	0.00000
GI_10092602-S	1400671	109.9	0.45155	4.419	58	94.5	0.68359	5.605	40	116.8	0.12393	6.864	42	113.3	0.54779
GI_10092603-S	2650605	113.0	0.37179	8.101	37	116.4	0.09558	8.898	48	111.0	0.25379	6.826	50	128.6	0.15030
GI_10092611-A	1660441	527.4	0.00000	18.695	46	412.1	0.00000	15.826	36	504.1	0.00000	24.501	40	785.4	0.00000
GI_10092616-S	5700086	185.0	0.00066	9.288	32	144.8	0.00198	9.256	39	140.9	0.00396	11.824	28	324.5	0.00000
GI_10092618-S	1050280	1747.8	0.00000	47.870	63	1429.9	0.00000	48.611	48	1504.4	0.00000	55.310	56	1272.0	0.00000
GI_10092638-S	4210093	202.7	0.00066	10.760	44	176.1	0.00000	8.350	33	193.3	0.00066	10.427	37	178.4	0.00066
GI_10092658-S	630114	139.3	0.02109	5.354	64	128.6	0.01648	8.054	48	144.8	0.00396	7.495	57	120.7	0.33158
GI_10092668-S	770403	119.6	0.21688	5.713	56	106.5	0.29334	6.749	54	113.7	0.18392	6.454	42	108.2	0.68688
GI_10092672-S	3800309	480.7	0.00000	25.498	38	346.5	0.00000	21.495	34	443.4	0.00000	16.815	40	852.9	0.00000
GI_10092690-S	1850647	196.6	0.00066	9.737	42	149.2	0.00198	9.297	56	175.8	0.00066	9.409	54	125.2	0.21490
GI_10190657-S	2100088	119.5	0.21753	4.872	62	97.0	0.60646	6.624	46	118.8	0.09888	6.607	52	126.9	0.17930
GI_10190665-I	5270324	101.8	0.66579	10.464	17	88.9	0.84443	5.056	25	95.4	0.78115	8.136	21	149.7	0.00923
GI_10190669-S	6620500	288.5	0.00000	14.900	38	231.1	0.00000	11.555	46	291.4	0.00000	12.755	35	116.6	0.45485
GI_10190671-S	6520093	126.9	0.10547	6.655	35	129.6	0.01516	10.390	35	113.7	0.18457	6.273	47	141.6	0.03362
GI_10190679-S	1570286	129.8	0.07251	8.419	38	107.3	0.27093	6.367	31	125.3	0.03889	8.888	31	129.1	0.13975
GI_10190681-S	1570435	136.3	0.03164	7.742	37	142.3	0.00264	8.304	53	133.5	0.01055	5.108	38	146.0	0.02044
GI_10190685-S	1690538	171.5	0.00132	9.528	39	147.0	0.00198	10.141	35	181.6	0.00066	9.539	48	120.0	0.35333
GI_10190687-S	4560575	120.1	0.20962	6.491	44	90.5	0.80751	6.662	40	95.7	0.77192	5.050	49	121.1	0.32103
GI_10190695-S	5900088	137.0	0.02835	10.089	29	120.1	0.05933	9.944	24	145.2	0.00396	16.438	15	157.6	0.00396

Rows=47312 Disp=47312 Sel=0 Filter=Filter is not active.

One set of observations (mean, se, detection etc) for each bead type. Local background correction was done and outliers removed before calculation of mean

All values are un-logged (1 - 2^{16})

The ‘beadarray’ Library

Collection of BeadArray analysis functions written using **R**

Functions for reading SAM and BeadChip data in bead summary or bead level format

Options for image processing

Also quality control, diagnostic checks and normalisation

Compatible with Bioconductor & R packages (e.g. *limma*, *affy)*

BIOCONDUCTOR

beadarray has been part of the Bioconductor project since December 2005

URL: http://www.bioconductor.org/packages/2.0/bioc/html/beadarray.html

Recently accepted for publication in Bioinformatics

Mark J. Dunning, Mike L. Smith, Matthew E. Ritchie, and Simon Tavaré
beadarray: R classes and methods for Illumina bead-based data
Bioinformatics Advance Access published on June 22, 2007.
doi:10.1093/bioinformatics/btm311

Why use beadarray?

Access to bead level data prior to processing by BeadStudio and re-visit image analysis

Quality control within arrays rather than just between arrays

Can be used to read expression / SNP / methylation and DASL data

Useful for those wishing to develop their own analysis methods (eg genotyping)

No need for Illumina (PC-based) analysis software

Reading bead level data

Reading bead level data into beadarray is as easy as running the following

```
> BLData = readIllumina()
```

This reads all the bead level files that it finds in the **R** working directory and estimates the foreground and background intensities for each bead on each array using the images

Setting `useImages=FALSE` will take the corrected intensities from the text files

Notes on `readIllumina`

Phenotypic information about the samples and metrics information provided by Illumina can also be read

`readIllumina` can read single or two-colour data from SAM or BeadChip experiments

Can take a lot of time and memory. Reading a BeadChip with image processing takes around 10 minutes and uses 2Gb RAM

Users can choose a smaller set of files to read, or choose not to repeat the image processing

What is BLData?

BLData is a BeadLevelList object, which is an *environment* object

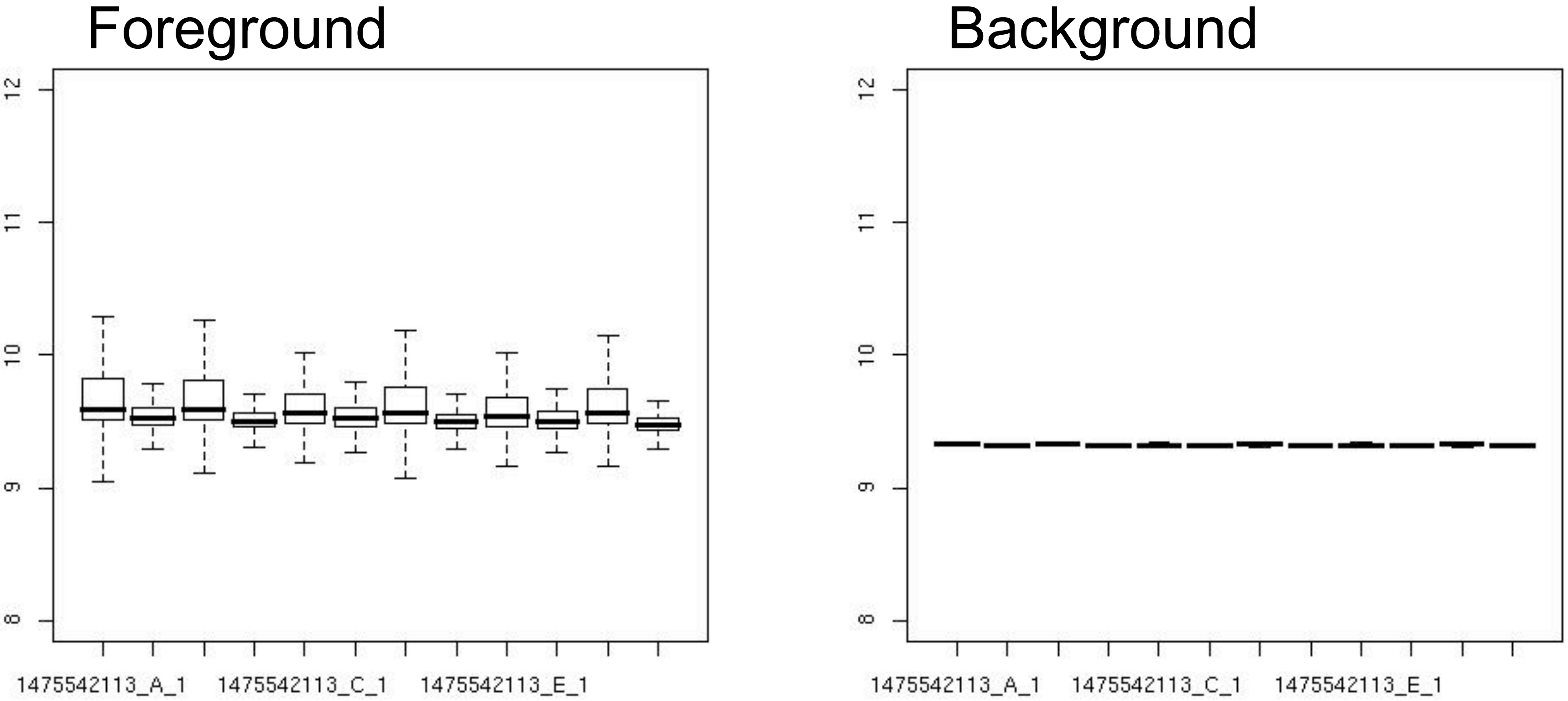
Information about BLData is organised into slots accessed by ‘@’ - beadData, arrayInfo

Arrays can be subset using ‘[[‘

The getArrayData function is also provided for convenience

See practical for examples

Raw Foreground and Background



Foreground

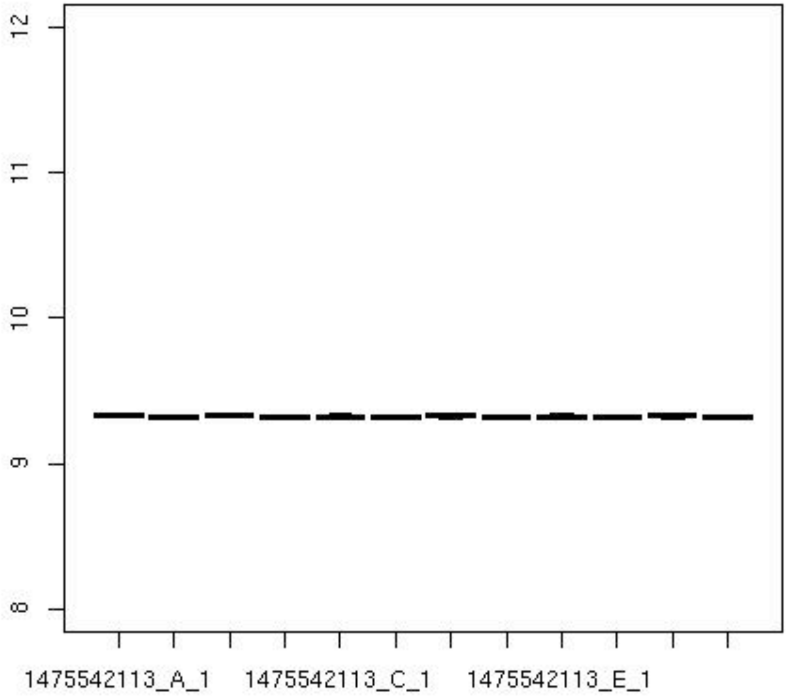
Background

Raw foreground and background intensities from each *strip* on a BeadChip (BeadStudio merges the strips together)

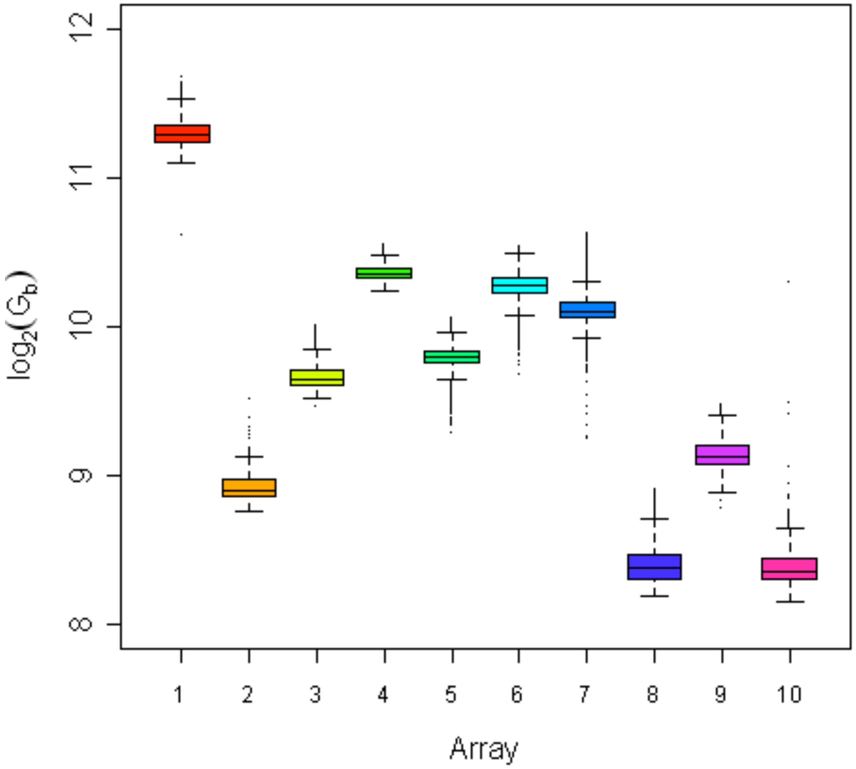
The different strips can have different properties

Compare with conventional arrays

Illumina



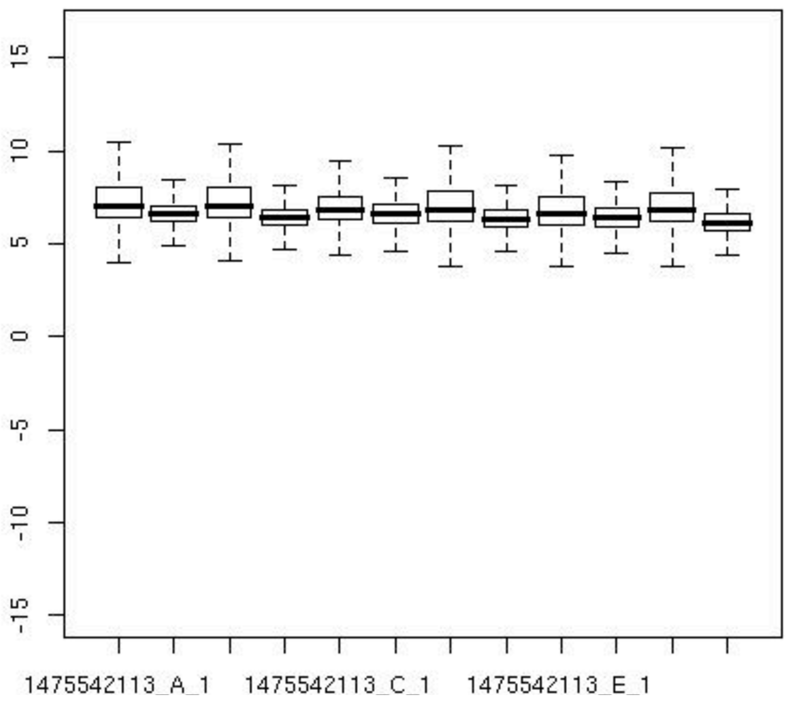
Two-Colour Array



Background Correction

beadarray includes all the background correction methods available in limma

The default option is to simply subtract the background from the foreground for each bead



Not as many negative values as for conventional arrays (<0.01% of beads are negative with Illumina data compared with 20-30%)

Spatial artefacts

Recall that spatial trends can be a cause for concern for microarray data

This should not be such a problem for BeadArrays due to the random positioning of beads and high number of replicates

beadarray includes functionality to check for serious spatial trends on arrays (as checking each array manually would be time-consuming)

The imageplot function can be used to investigate spatial trends (see practical)

imageplot

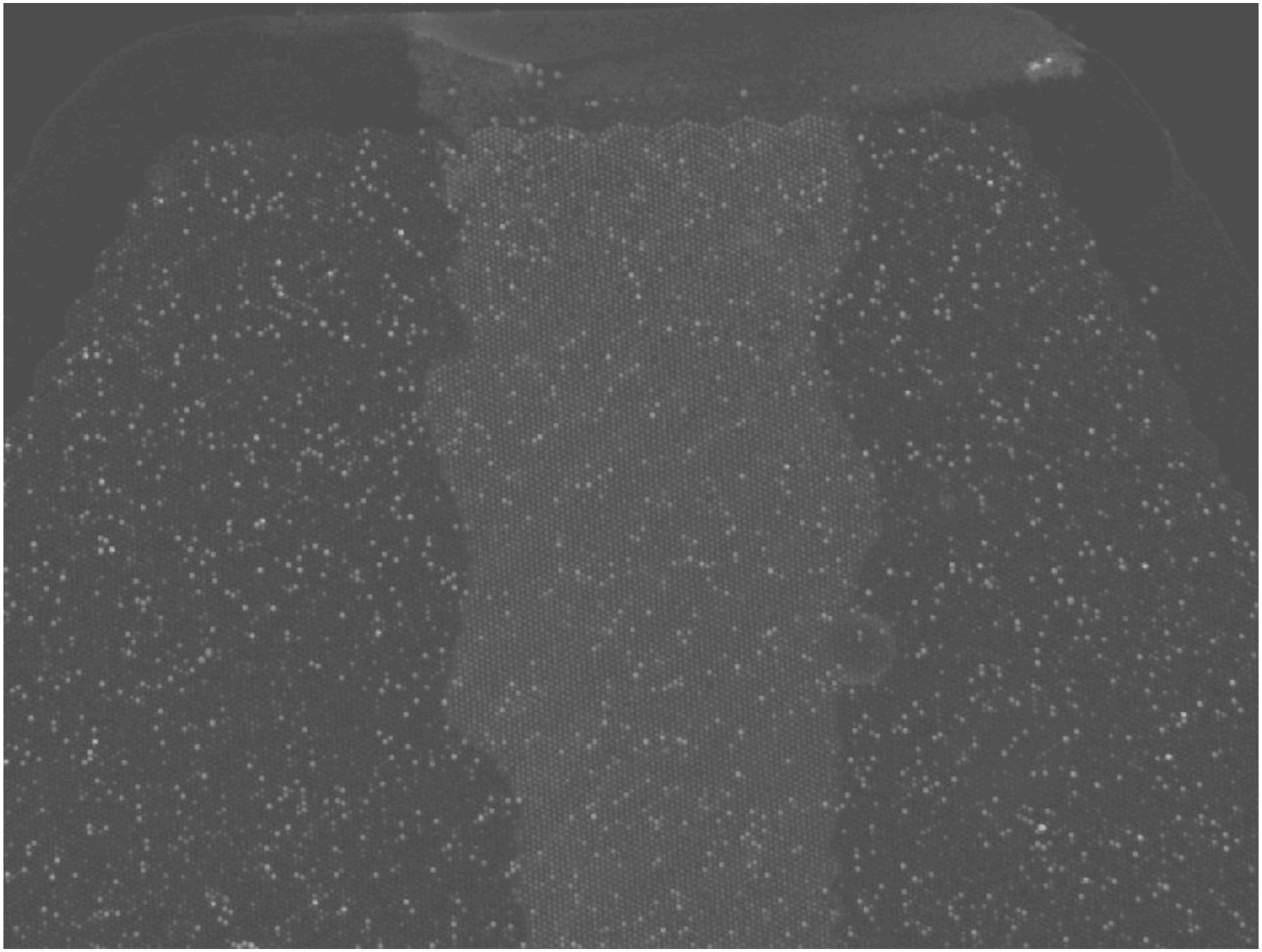
```
>imageplot(BLData, what="G")
```

Useful things to plot include foreground, background, residuals

Spatial artefacts will often associate with outliers

This can give more detailed diagnostics for particular arrays

All of which is not possible with summarised data

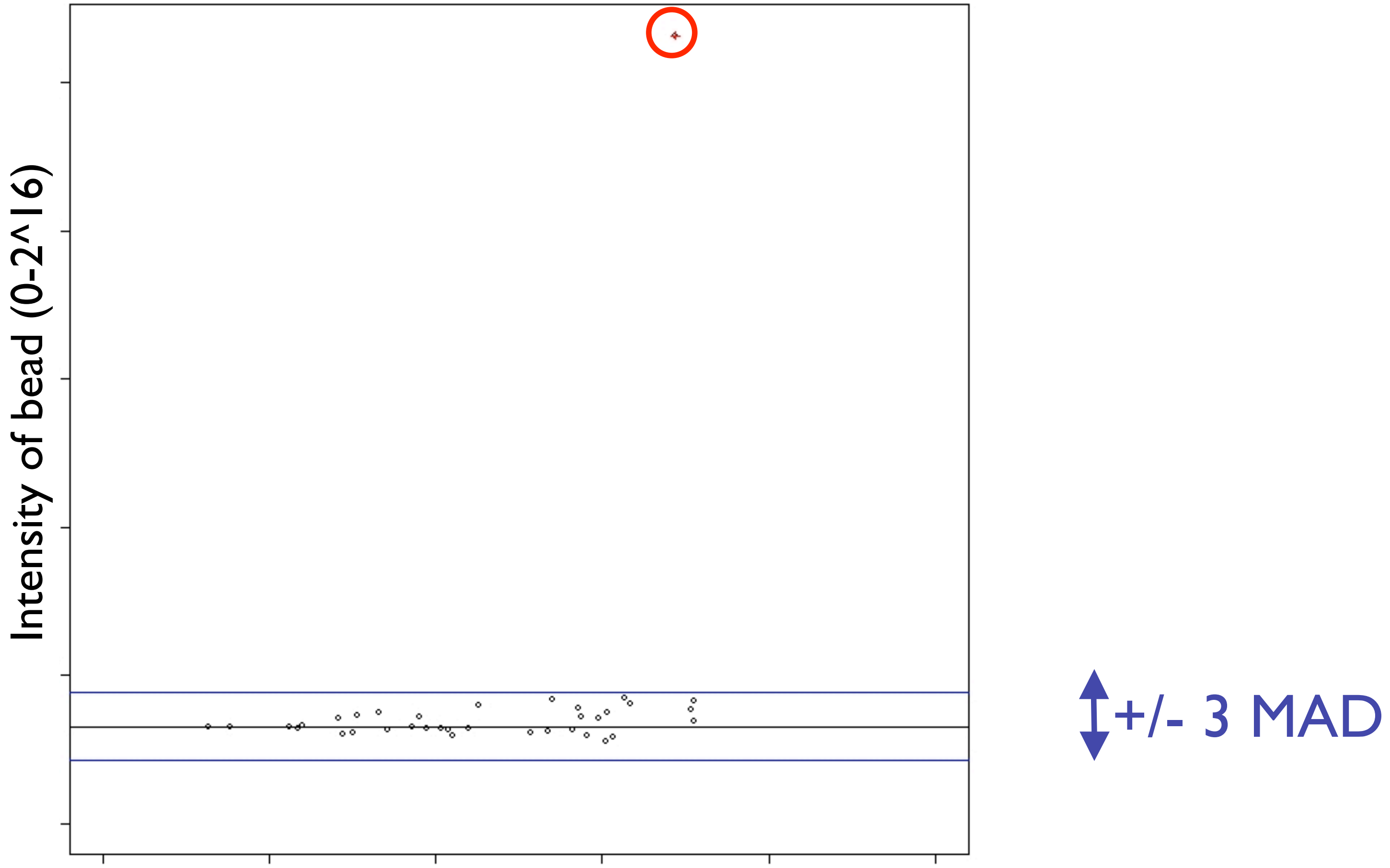


Artefacts can be seen on original images with some effort

Creating Bead Summary Data

We use the Illumina method to remove outliers using a 3 median absolute deviation cut-off from the median for each bead type

```
>BSData = createBeadSummaryData(BLData)
```



Remarks

Can choose to summarise the data on the log2 scale

The resulting object BSData is an ExpressionSetIllumina object which extends an ExpressionSet. The expression matrix can be easily extracted for further analysis

If two-colour data is given, the two channels are summarised separately to give a SnpSetIllumina object

Storing bead summary data

`BSData` is now an `ExpressionSetIllumina` object sharing many common properties with other Bioconductor objects

The expression values can be accessed using the `exprs` function

```
> E=exprs(BSData)
> dim(E)
[1] 47293    18
> E[1:3,1:3]
              AVG_Signal.IH.1 AVG_Signal.IC.1 AVG_Signal.IH.2
GI_10047089-S            87.8           131.8           231.9
GI_10047091-S           161.8           130.8           258.6
GI_10047093-S           481.2           401.4           499.4
```

For more details see the practical…

readBeadSummaryData

```
>BSData = readBeadSummaryData(dataFile, qcInfo,
sampleSheet)
```

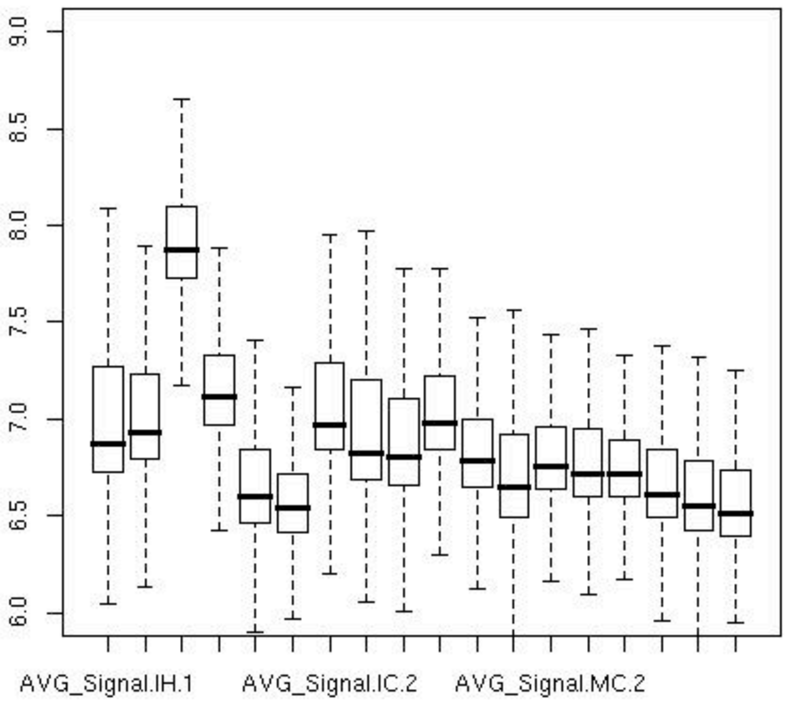
Warning: May need to change skip and sep parameters depending on version of BeadStudio

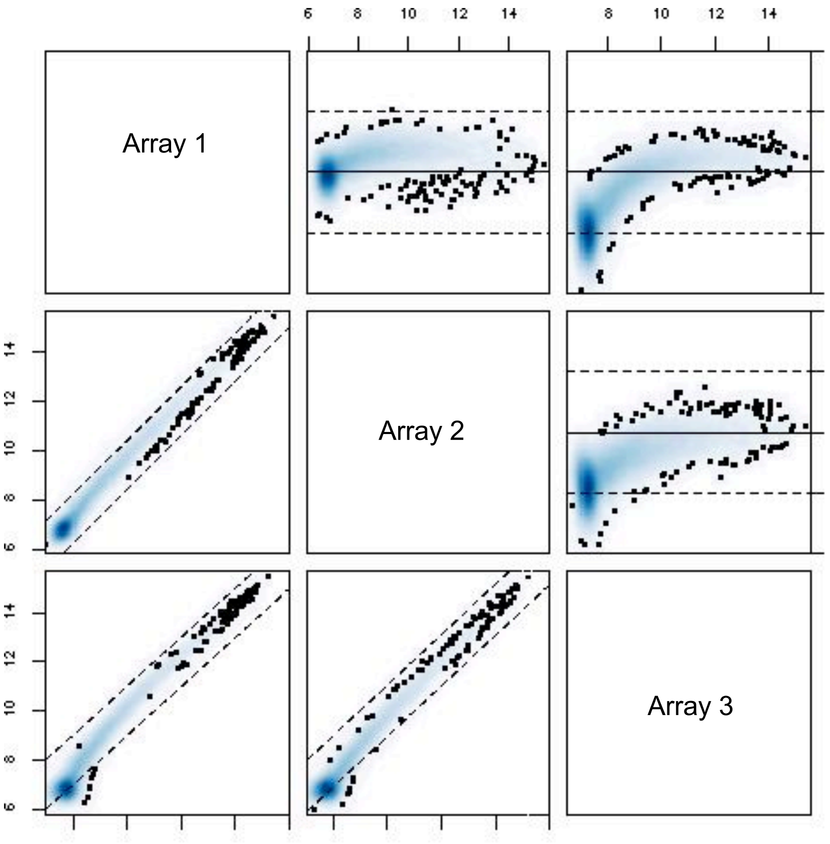
Eg

```
>BSData = readBeadSummaryData(dataFile, qcInfo,
sampleSheet, skip=7, sep=",")
```

Also, column headings in BeadStudio output sometimes change (BEAD_STDEV -> BEAD_STDERR)

Quality Control





```
> boxplot(as.data.frame(log2(E)),outline=FALSE, ylim=c(6,9))


> plotMAXY(E, arrays=1:3)
```

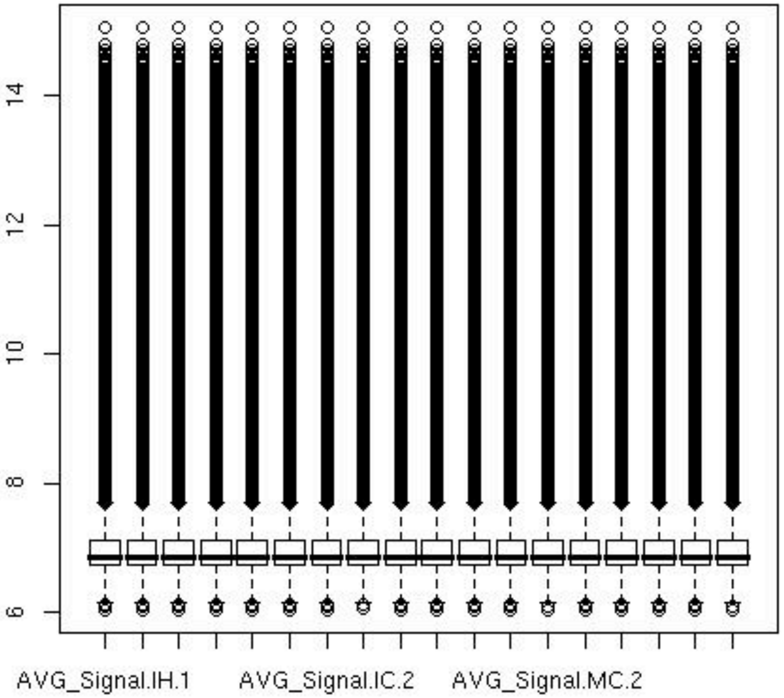
Normalisation

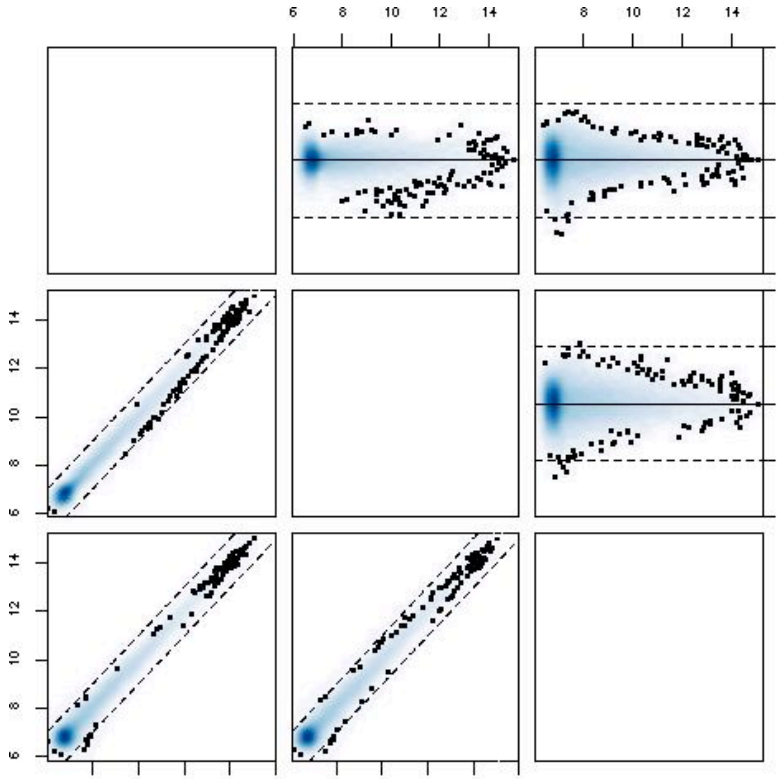
Illumina data seems to be of good quality, however some trends can still be seen in the data (eg decrease in intensity across a chip)

Important not to remove any biological effects

Quantile normalisation seems to be effective

Or any other normalisation method from Bioconductor which can be used on an expression matrix. Many can be found in the affy package





```
>E = normaliseIllumina(BSData, method="quantile", transform="log2")
>boxplot(as.data.frame(log2(E)), outline=TRUE)
>plotMAXY(E, arrays=1:3)
```

And now over to the practical….

Acknowledgements

Computational Biology Group (Cambridge)

Matthew Ritchie

Mike Smith

Julie Addison

Natalie Thorne

Andy Lynch

Nuno Barbosa-Morais

Simon Tavaré

Dermitzakis group (Sanger Institute - Cambridge)

Manolis Dermitzakis

Barbara Stranger

Matthew Forrest

Caldas Lab

Inma Spiteri

Anna Git

Illumina (San Diego)

Brenda Kahl

Semyon Kruglyak

Gary Nunn

UCSD(San Diego)

Roman Sasik

CANCER RESEARCH UK

MRC Medical Research Council