QC and Affymetrix data

[1]Claire Wilson, [2]Stuart D Pepper, [1]Crispin J Miller
[1]Bioinformatics Group, [2]Moloecular Biology Core Facility
Paterson Institute for Cancer Research, Christie Hospital NHS Trust, Wilmslow Road, Withington, Manchester M20 4BX UK.

Quality Control (QC) assessment is a crucial first step in successful data analysis: before any comparisons can be performed it is necessary to check that there were no problems with sample processing, and that arrays are of sufficient quality to be included in a study. The Affymetrix platform has a collection of QC metrics and accompanying guidelines that aid the identification of problematic arrays. Detailed information about these can be found in the Affymetrix 'Data Analysis Fundamentals Manual' (located at http://www.affymetrix.com). Here, we provide a summary (and our own interpretation) of these metrics, and show how the QC functions within the `simpleaffy` package in BioConductor can be used to assess array quality. `Simpleaffy` provides high level functions for reading Affymetrix .CEL files, phenotypic data, and then computing simple things from them, such as t-tests, fold changes etc. It makes heavy use of the `affy` package within BioConductor and also has some basic scatter plot functions and mechanisms for generating high resolution journal figures.

The example dataset

The data used for all the following examples is taken from Wilson et al. (2004) 'Amplification protocols introduce systematic but reproducible errors into gene expression studies' (Biotechniques 36(3):498-506), and can be downloaded from http://bioinformatics.picr.man.ac.uk. The experiment investigated how gene expression measurements changed when two different (but related) protocols from Affymetrix were used to prepare the RNA, and also how the data varied with different amounts of starting material. These data are useful here, because the experiment was, in part, designed to investigate the limits of the small sample protocol, and as a result, provides us with a subset of arrays that failed QC.

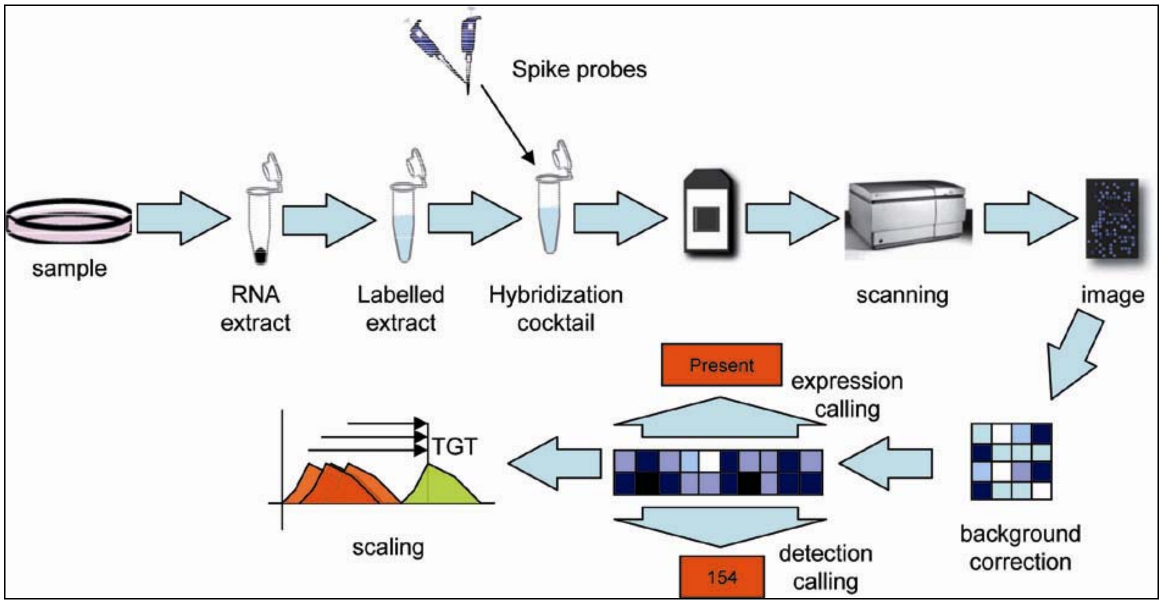
RNA from two cell lines (MCF7 and MCF10A) was extracted with replication to provide 3 large pools of RNA for each cell line. Aliquots from these were taken and used for subsequent processing. The 'cell.line' column specifies what cell line was used; the 'rna' column specifies the amount of RNA processed. All arrays produced from 10μg of starting material were processed using the standard protocol (http://bioinformatics.picr.man.ac.uk/mbcf/downloads/GeneChip_Target_Prep_Protocol-CR-UK_v3.pdf); those with 1, 10 or 100ng used the small sample protocol (http://bioinformatics.picr.man.ac.uk/mbcf/downloads/GeneChip_Small_Sample_Target_Preparation_Protocol-CR-UK_v1.pdf). A description of the .CEL filenames and the RNA that was hybridised to them is shown in the table below, and forms the basis of the 'covdesc' file used by simpleaffy (a white-space delimited file that describes what samples went on what chips).

Table I: Description of the samples analysed in the amplification experiment

.CEL file	cell.line	rna
MCF10A_r1.CEL	mcf10a	10ug
MCF10A_r2.CEL	mcf10a	10ug

MCF10A_r3.CEL	mcf10a	10ug
MCF7_r1.CEL	mcf7	10ug
MCF7_r2.CEL	mcf7	10ug
MCF7_r3.CEL	mcf7	10ug
a100MCF10A_r1.CEL	mcf10a	100ng
a100MCF10A_r2.CEL	mcf10a	100ng
a100MCF10A_r3.CEL	mcf10a	100ng
a100MCF7_r1.CEL	mcf7	100ng
a100MCF7_r2.CEL	mcf7	100ng
a100MCF7_r3.CEL	mcf7	100ng
a10MCF10A_r1.CEL	mcf10a	10ng
a10MCF10A_r2.CEL	mcf10a	10ng
a10MCF10A_r3.CEL	mcf10a	10ng
a10MCF7_r1.CEL	mcf7	10ng
a10MCF7_r2.CEL	mcf7	10ng
a10MCF7_r3.CEL	mcf7	10ng
a1MCF10A_r1.CEL	mcf10a	1ng
a1MCF10A_r2_2.CEL	mcf10a	1ng
a1MCF10A_r3.CEL	mcf10a	1ng
a1MCF7_r1_2.CEL	mcf7	1ng
a1MCF7_r2.CEL	mcf7	1ng
a1MCF7_r3_2.CEL	mcf7	1ng

The QC metrics implemented in `simpleaffy`



The figure above shows the steps one might take processing a set of samples before analysing the data using Affymetrix's standard algorithms. It is important to be familiar with this process, not only because the QC metrics are designed to identify issues that occur at different stages of the process, but also because many of them are based around values calculated by the MAS 5.0 algorithms. Details of how these algorithms work can be found in Affymetrix's 'Data Analysis Fundamentals Manual' (located at http://www.affymetrix.com), and at http://bioinformatics.picr.man.ac.uk.

The `simpleaffy` function, `qc`, generates the most commonly used metrics:

1. Average background
2. Scale factor
3. Number of genes called present
4. 3' to 5' ratios for β-actin and GAPDH
5. Values for spike-in control transcripts

All of these values are parameters computed for/from the MAS 5.0 algorithm. The standard recommendations from Affymetrix are as follows:

Average background
This should be similar across all chips, if chips have significantly different average backgrounds this could be for a number of reasons. It might be simply that the overall signal from the array is greater, perhaps because different amounts of cRNA were present in the hybridisation cocktails, or because the hybridisation was more efficient in one of the reactions, incorporating more label, and producing a brighter chip.

Scale factors
The default normalisation used by MAS 5.0 (and many other algorithms) makes the assumption that gene expression does not change significantly for the vast majority of transcripts in an experiment. (Note that this assumption is also explicit in any analysis that looks for a relatively small number of changing genes within a transcript population containing many thousands (for example, looking for ~200 differentially expressed probesets from the ~54,000 found on the U133 plus 2 array)).

One consequence of this is that the trimmed mean intensity for each array should be constant., and by default, MAS 5.0 scales the intensity for every sample so that each array has the same mean. The amount of scaling applied is represented by the 'scale factor', which, therefore, provides a measure of the overall expression level for an array, and (assuming all else remains constant), a reflection of how much labelled RNA is hybridised to the chip. Large variations in scale factors signal cases where the normalisation assumptions are likely to fail due to issues with sample quality or amount of starting material. Alternatively, they might occur if there have been significant issues with RNA extraction, labelling, scanning or array manufacture. In order to successfully compare data produced using different chips, Affymetrix recommend that their scale factors should be within 3-fold of one another.

Number of genes called present (% Present)
Present/Marginal/Absent calls are generated by looking at the difference between PM and MM values for each probe pair in a probeset. Probesets are flagged Marginal or Absent when the PM values for that probeset are not considered to be significantly above the MM probes. As with scale factors, large differences between the numbers of genes called present on different arrays can occur when varying amounts of labelled RNA have been successfully hybridized to the chips. This can occur for similar reasons (differences in array processing pipelines, variations in the amount of starting material, etc.). The '% Present' call simply represents the percentage of probesets called Present on an array. As with Scale Factors, significant *variations* in % Present call across the arrays in a study should be treated with caution. Note that the absolute value is generally not a good metric – some cells naturally express more genes than others.

3' to 5' ratios
Most cell types ubiquitously express β-actin and GAPDH. These are relatively long genes, and the majority of Affymetrix chips contain separate probesets targeting the 5', mid and 3' regions of their transcripts. By comparing the amount of signal from the 3' probeset to either the mid or 5' probesets, it is possible to obtain a measure of the quality of the RNA hybridised to the chip. If the ratios are high then this indicates the presence of truncated transcripts. This may occur if the *in vitro* transcription step has not performed well or if there is general degradation of the RNA. Hence, the ratio of the 3' and 5' signal gives a measure of RNA quality. If RNA has been prepared using the Affymetrix Small Sample protocol instead of the Affymetrix Standard Protocol, it is recommended that the 3' to mid ratios be used. This is because the extra amplification step within the small sample protocol is likely to increase the frequency of short transcripts in solution and unavoidably introduce some 3' bias into the population of labelled transcripts.

GAPDH is the smaller of the two genes and the 3':5' ratio should always be at or around 1. In the examples below, we have set a threshold of 1.25, based on our own experiences. It is frequently observed at just under 1; this should not be taken to mean that there is more of the 5' probeset than the 3' probeset. Affymetrix suggest that a β-actin 3':5' ratio of less than 3 is acceptable. Because of the inherent 3' bias in the transcript population, it is known that data quality is not significantly affected when 3':5' or 3':mid ratios fall within these bounds.

Spike-in probesets (Hybridisation controls)
In order to verify the efficiency of the hybridisation step, some additional labelled cRNAs are added during the latter stages of the sample preparation protocol. These transcripts (BioB, BioC, BioD and CreX) are derived from *Bacillus subtiliis*: nothing else within the hybridisation cocktail should bind to their probesets. They are spiked into the solution just prior to it being placed on an array, and their intensity is dependent on the hybridisation/scanning steps. BioB is added at a concentration of 1.5pM, corresponding to approximately three transcripts per cell, the lower limit of detection for the system. BioC, BioD and CreX are spiked in at increasing concentrations. Ideally, BioB should be called present on every array: with the newer arrays and protocols we have found this to be the case, however an acceptable level is for it to be called present on 70% of the chips in an experiment. If BioB is routinely absent then the assay is performing with suboptimal sensitivity.

Assessing QC measures using `simpleaffy`

1. Calling `qc` function

The function `qc` within `simpleaffy` analyses either normalised expression data, or raw data held within an AffyBatch object. It produces an object of class `QCStats` that contains QC metrics for each array in a project.

```
# Load the simpleaffy library
> library (simpleaffy)
 # view the help pages for the function qc
> ?qc
```

NOTE: The function can be called with raw data and normalised data (an AffyBatch object) ONLY if the normalized data was generated using `call.exprs` and the `mas5` method. This is because `call.exprs` generates a list of scale factors for each chip and these are stored within the AffyBatch's `description@preprocessing` slot. Scale factors are automatically written out to screen when `call.exprs` is applied using `mas5`.

```
# Read in all the .CEL files in the current directory and attach a phenoData object
# covdesc describes the samples and is used to create the phenoData object
# for more information type ?read.affy
> ampli.data <- read.affy(”covdesc”)
# Normalise the data using call.exprs and mas5.
# All chips will be scaled so that their mean intensity is 100
# For more information type ?call.exprs
> ampli.eset <- call.exprs(ampli.data,"mas5")
Background correcting
Retrieving data from AffyBatch...done.
Computing expression calls...
........................done.
scaling to a TGT of 100 ...Scale factor for: ./MCF10A_r1.CEL 0.420253004956145
Scale factor for: ./MCF10A_r2.CEL 0.578922235057308
Scale factor for: ./MCF10A_r3.CEL 0.394833649674273
Scale factor for: ./MCF7_r1.CEL 0.291660288551772
Scale factor for: ./MCF7_r2.CEL 0.347797229427576
Scale factor for: ./MCF7_r3.CEL 0.318539156223863
Scale factor for: ./a100MCF10A_r1.CEL 0.451200408149428
Scale factor for: ./a100MCF10A_r2.CEL 0.374087365216242
Scale factor for: ./a100MCF10A_r3.CEL 0.487207458186858
Scale factor for: ./a100MCF7_r1.CEL 0.287589038443532
Scale factor for: ./a100MCF7_r2.CEL 0.284974495606986
Scale factor for: ./a100MCF7_r3.CEL 0.376484876958665
Scale factor for: ./a10MCF10A_r1.CEL 0.945369717387057
Scale factor for: ./a10MCF10A_r2.CEL 1.96143998371977
Scale factor for: ./a10MCF10A_r3.CEL 0.841535921004799
Scale factor for: ./a10MCF7_r1.CEL 0.38546207789797
Scale factor for: ./a10MCF7_r2.CEL 0.413217979099816
Scale factor for: ./a10MCF7_r3.CEL 0.48270303222115
Scale factor for: ./a1MCF10A_r1.CEL 0.923881708561097
Scale factor for: ./a1MCF10A_r2_2.CEL 2.29265382455517
Scale factor for: ./a1MCF10A_r3.CEL 4.02474782730734
Scale factor for: ./a1MCF7_r1_2.CEL 3.59230194412852
Scale factor for: ./a1MCF7_r2.CEL 3.1613079022588
Scale factor for: ./a1MCF7_r3_2.CEL 2.01330394340637
Getting probe level data...
Computing p-values
Doing PMA Calls

# see what data is stored in ampli.eset@description@preprocessing
> names(ampli.eset@description@preprocessing)
[1] "filenames"   "affyversion" "sfs"         "tgt"

# access each piece of information within ampli.eset@description@preprocessing
# scale factors
> ampli.eset@description@preprocessing$sfs
 [1] 0.4202530 0.5789222 0.3948336 0.2916603 0.3477972 0.3185392 0.4512004
 [8] 0.3740874 0.4872075 0.2875890 0.2849745 0.3764849 0.9453697 1.9614400
[15] 0.8415359 0.3854621 0.4132180 0.4827030 0.9238817 2.2926538 4.0247478
[22] 3.5923019 3.1613079 2.0133039
# filenames so that the scale factors can be related to their chips
> ampli.eset@description@preprocessing$filenames
 [1] "./MCF10A_r1.CEL"     "./MCF10A_r2.CEL"     "./MCF10A_r3.CEL"
 [4] "./MCF7_r1.CEL"       "./MCF7_r2.CEL"       "./MCF7_r3.CEL"
 [7] "./a100MCF10A_r1.CEL" "./a100MCF10A_r2.CEL" "./a100MCF10A_r3.CEL"
[10] "./a100MCF7_r1.CEL"   "./a100MCF7_r2.CEL"   "./a100MCF7_r3.CEL"
[13] "./a10MCF10A_r1.CEL"  "./a10MCF10A_r2.CEL"  "./a10MCF10A_r3.CEL"
[16] "./a10MCF7_r1.CEL"    "./a10MCF7_r2.CEL"    "./a10MCF7_r3.CEL"
[19] "./a1MCF10A_r1.CEL"   "./a1MCF10A_r2_2.CEL" "./a1MCF10A_r3.CEL"
[22] "./a1MCF7_r1_2.CEL"   "./a1MCF7_r2.CEL"     "./a1MCF7_r3_2.CEL"
# tgt is the target intensity each chip was scaled to
> ampli.eset@description@preprocessing$tgt
[1] 100
# which version of the affy package was used
> ampli.eset@description@preprocessing$affyversion
[1] "1.4.30"
> qc.data <- qc(ampli.data, ampli.eset)
```

Alternatively, you can call `qc` with just the raw data, in which case it will be normalized using `mas5` so that the scale factors can be calculated. This will obviously take a little more time to run.

2. Data stored within the object created by calling `qc()`

`qc` returns an object containing scale-factors, % present, average, minimum, maximum and mean background intensities, and bioB, bioC, bioD and creX present calls (1=present; 0=not present). It also stores 3', 5' and mid values for the QC probes, `ratios(qc)` generates a table of qc ratios for these probes. See `?qc` for more details.

```
> slotNames(qc.data)
[1] "scale.factors"      "target"             "percent.present"
[4] "average.background" "minimum.background" "maximum.background"
[7] "spikes"             "qc.probes"          "bioBCalls"

# scale.factors contains a list of scale factors applied to each chip;
> qc.data@scale.factors
 [1] 0.4202530 0.5789222 0.3948336 0.2916603 0.3477972 0.3185392 0.4512004
 [8] 0.3740874 0.4872075 0.2875890 0.2849745 0.3764849 0.9453697 1.9614400
[15] 0.8415359 0.3854621 0.4132180 0.4827030 0.9238817 2.2926538 4.0247478
[22] 3.5923019 3.1613079 2.0133039

# target is the target intensity that each chip was scaled to;
> qc.data@target
[1] 100

# percent.present is a list of the percentage of probesets called present on each chip;
> qc.data@percent.present
     ./MCF10A_r1.CEL.present     ./MCF10A_r2.CEL.present     ./MCF10A_r3.CEL.present
                    53.22892                    51.65373                    52.91029
       ./MCF7_r1.CEL.present       ./MCF7_r2.CEL.present       ./MCF7_r3.CEL.present
                    57.24543                    56.10106                    55.37405
 ./a100MCF10A_r1.CEL.present ./a100MCF10A_r2.CEL.present ./a100MCF10A_r3.CEL.present
                    47.58785                    48.04560                    48.35076
   ./a100MCF7_r1.CEL.present   ./a100MCF7_r2.CEL.present   ./a100MCF7_r3.CEL.present
                    50.53628                    52.21469                    49.03738
  ./a10MCF10A_r1.CEL.present  ./a10MCF10A_r2.CEL.present  ./a10MCF10A_r3.CEL.present
                    41.87048                    33.24059                    40.26837
    ./a10MCF7_r1.CEL.present    ./a10MCF7_r2.CEL.present    ./a10MCF7_r3.CEL.present
                    48.12189                    48.64695                    48.58861
   ./a1MCF10A_r1.CEL.present ./a1MCF10A_r2_2.CEL.present   ./a1MCF10A_r3.CEL.present
                    41.40825                    33.62653                    27.55913
   ./a1MCF7_r1_2.CEL.present     ./a1MCF7_r2.CEL.present   ./a1MCF7_r3_2.CEL.present
                    32.76938                    31.13136                    35.28699

# average.background, minimum.background, maximum.background are all lists detailing
# the average, minimum and maximum background for each chip;
```

```
> qc.data@average.background
    ./MCF10A_r1.CEL     ./MCF10A_r2.CEL     ./MCF10A_r3.CEL       ./MCF7_r1.CEL
           63.33604            59.67562            63.08459            54.79780
      ./MCF7_r2.CEL       ./MCF7_r3.CEL ./a100MCF10A_r1.CEL ./a100MCF10A_r2.CEL
           56.55383            59.26790            82.39357            80.43839
./a100MCF10A_r3.CEL   ./a100MCF7_r1.CEL   ./a100MCF7_r2.CEL   ./a100MCF7_r3.CEL
           72.82438            82.97151            80.14180            82.91695
 ./a10MCF10A_r1.CEL  ./a10MCF10A_r2.CEL  ./a10MCF10A_r3.CEL    ./a10MCF7_r1.CEL
           64.83461            61.27757           116.11719            77.48265
   ./a10MCF7_r2.CEL    ./a10MCF7_r3.CEL   ./a1MCF10A_r1.CEL ./a1MCF10A_r2_2.CEL
           76.41997            75.32842            81.97728            78.91661
  ./a1MCF10A_r3.CEL   ./a1MCF7_r1_2.CEL     ./a1MCF7_r2.CEL   ./a1MCF7_r3_2.CEL
           68.31636            59.52758            81.64647            93.72851
> qc.data@minimum.background
    ./MCF10A_r1.CEL     ./MCF10A_r2.CEL     ./MCF10A_r3.CEL       ./MCF7_r1.CEL
           58.56850            56.92324            58.85309            51.95272
      ./MCF7_r2.CEL       ./MCF7_r3.CEL ./a100MCF10A_r1.CEL ./a100MCF10A_r2.CEL
           53.21018            54.04524            77.05848            70.90370
./a100MCF10A_r3.CEL   ./a100MCF7_r1.CEL   ./a100MCF7_r2.CEL   ./a100MCF7_r3.CEL
           69.39079            75.62456            74.40889            70.72456
 ./a10MCF10A_r1.CEL  ./a10MCF10A_r2.CEL  ./a10MCF10A_r3.CEL    ./a10MCF7_r1.CEL
           61.17076            58.02036            95.00356            71.73734
   ./a10MCF7_r2.CEL    ./a10MCF7_r3.CEL   ./a1MCF10A_r1.CEL ./a1MCF10A_r2_2.CEL
           70.98562            71.13894            76.71987            70.66398
  ./a1MCF10A_r3.CEL   ./a1MCF7_r1_2.CEL     ./a1MCF7_r2.CEL   ./a1MCF7_r3_2.CEL
           61.93977            55.82989            75.88498            84.54410
> qc.data@maximum.background
    ./MCF10A_r1.CEL     ./MCF10A_r2.CEL     ./MCF10A_r3.CEL       ./MCF7_r1.CEL
           68.30370            62.90242            67.32432            56.89239
      ./MCF7_r2.CEL       ./MCF7_r3.CEL ./a100MCF10A_r1.CEL ./a100MCF10A_r2.CEL
           59.84686            64.49775            87.14364            86.77520
./a100MCF10A_r3.CEL   ./a100MCF7_r1.CEL   ./a100MCF7_r2.CEL   ./a100MCF7_r3.CEL
           76.66844            89.24477            86.83688            97.07952
 ./a10MCF10A_r1.CEL  ./a10MCF10A_r2.CEL  ./a10MCF10A_r3.CEL    ./a10MCF7_r1.CEL
           67.90226            64.38952           135.87939            80.52142
   ./a10MCF7_r2.CEL    ./a10MCF7_r3.CEL   ./a1MCF10A_r1.CEL ./a1MCF10A_r2_2.CEL
           81.11707            78.84590            90.65008            87.51111
  ./a1MCF10A_r3.CEL   ./a1MCF7_r1_2.CEL     ./a1MCF7_r2.CEL   ./a1MCF7_r3_2.CEL
           79.99243            62.54194            87.70483           103.33656

# spikes is a matrix containing normalised values for each of the spike controls
> colnames(qc.data@spikes)
[1] "AFFX-r2-Ec-bioB-3_at" "AFFX-r2-Ec-bioC-3_at" "AFFX-r2-Ec-bioD-3_at"
[4] "AFFX-r2-P1-cre-3_at"
```

A note on GAPDH and β-actin probesets

For some arrays, more than one probeset targets the GAPDH and β-actin transcripts. In this situation, we've attempted to make a sensible choice as to which probeset to use in calculating 3':5' ratios. To find out which probesets are used for your arrays use the following methods on the AffyBatch object that contains the raw data and the syntax described below:
`getGapdh3`, `getGapdhM`, `getGapdh5`, `getActinb3`, `getActinbM`, `getActinb5`, `getBioB`, `getBioC`, `getBioD`, `getCreX`.
e.g. to find out which probeset best represents the GAPDH 3' region:

```
> getGapdh3(cleancdfname(cdfName(ampli.data)))
```

A note of warning!

The qc function has not been tested on every possible chip type – the majority of development has been on HGU95A arrays and newer. Again, for a list of which chips it has been tested on, how the testing was done, and the results of the comparisons, see the simpleaffy website (http://bioinformatics.picr.man.ac.uk/simpleaffy/index.jsp).

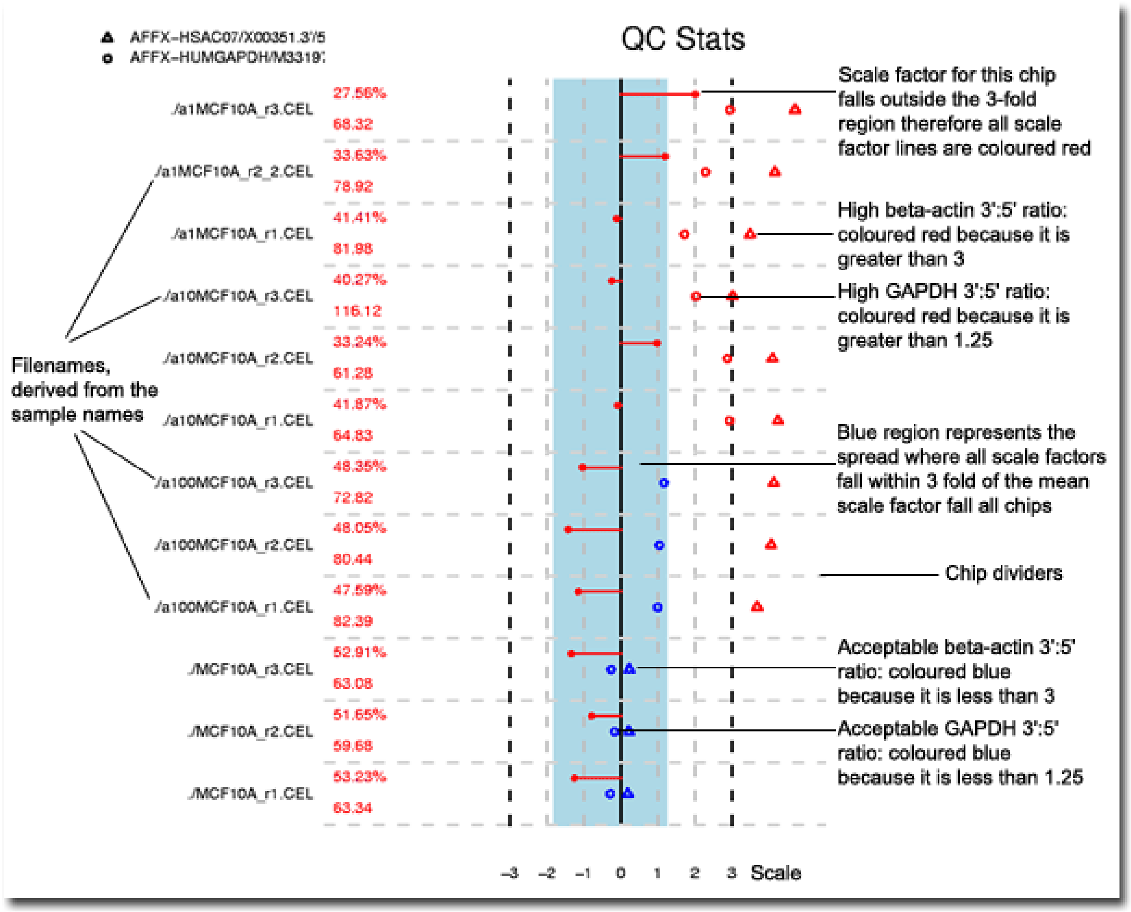
3. Visualising data stored in a `QCStats` object

A plot of qc data is obtained by `plot(qc.data)`. The resultant image takes a bit of explaining, but the data is portrayed in a way that makes outlier chips easily identifiable. In the following examples a subset of the amplification dataset is first used to illustrate the different features of the qc plot, before we consider all the qc data for the amplification dataset. Finally, ways to customise the qc plot are described.

```
# generate a subset of the amplification data
# contains only data on MCF10A cells
> subset <- ampli.data[,c(1:3,7:9,13:15,19:21)]
> subset.eset <- call.exprs(subset, “mas5”)
> qc.subset <- qc(subset, subset.eset)

# plot the qc graph
> plot(qc.subset)
# to view options for plotting the qc plot
> ?plot.qc.stats

# to save the diagram in a .png format use the journalpng function
# view help files for journalpng function
> ?journalpng
> journalpng(file=”qc_subset.png”, height=6, width=6)
> plot(qc.subset)
> dev.off()
```



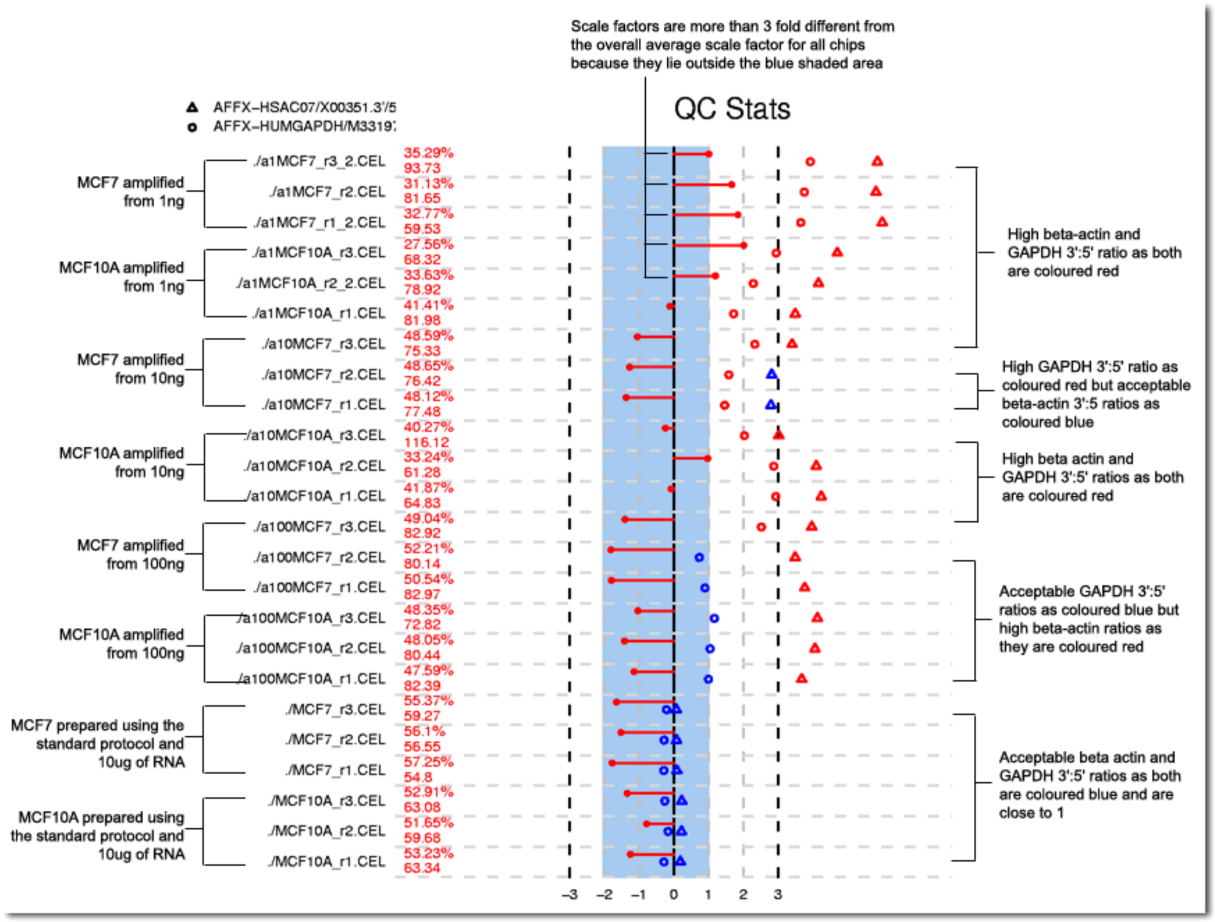
The figure is plotted from the bottom up with the first chip being at the base of the diagram and the last chip in the QCStats object at the top. If the standard steps for generating a QCStats object are followed, then this corresponds to the order of your samples in the AffyBatch object. Dotted horizontal lines separate the plot into rows, one for each chip. Dotted vertical lines provide a scale from -3 to 3.

Each row shows the %present, average background, scale factors and GAPDH / β-actin ratios for an individual chip.

- GAPDH 3':5' values are plotted as circles. According to Affymetrix they should be about 1. GAPDH values that are considered potential outliers (ratio > 1.25) are coloured red, otherwise they are blue.
- β-actin, 3':5' ratios are plotted as triangles. Because this is a longer gene, the recommendation is for the 3':5' ratios to be below 3; values below 3 are coloured blue, those above, red.
- The blue stripe in the image represents the range where scale factors are within 3-fold of the mean for all chips. Scale factors are plotted as a line from the centre line of the image. A line to the left corresponds to a down-scaling, to the right, to an up-scaling. If any scale factors fall outside this '3-fold region', they are all coloured red, otherwise they are blue.
- %present and average background, are listed to left of the figure. These are discussed in more detail below.

4. Analysing QC data for the amplification experiment

```
# R code used to generate the data
# Read in the data, for more information type ?read.affy
> ampli.data <- read.affy(”covdesc”)
# Normalise the data using call.exprs and mas5.
# All chips will be scaled so that their mean intensity is 100
# For more information type ?call.exprs
> ampli.eset <- call.exprs(ampli.data,"mas5")
> qc.data <- qc(ampli.eset, ampli.data)
> plot(qc.data)
```



In order to assess the quality of data generated in this experiment, we will consider four out of the five metrics mentioned at the start.

1. Average background

The average background for each chip shows a considerable amount of variation. This is represented in the figure above by colouring the average background values for all chips red. However upon closer inspection it is clear that the average background is higher for those samples that were processed using the amplification protocol compared to those samples processed using the standard protocol. In particular the average background for the 3rd replicate in the ‘MCF10A amplified from 10ng’ set has a value around 116, while all other values tend to be between 55 and 95. Although this may indicate that there is a problem with this sample all other QC measures correlate well with those of its peers (as seen below).

2. Scale factor

The scale factors for all ‘MCF7 amplified from 1ng’ replicates and replicates 2 and 3 of the ‘MCF10A amplified from 1ng’ samples are all greater than 3-fold away from the average scale factor for all samples. This suggests that the overall intensity of these samples was lower, resulting in higher scale factors. This in turn may result from there being less RNA present in these samples.

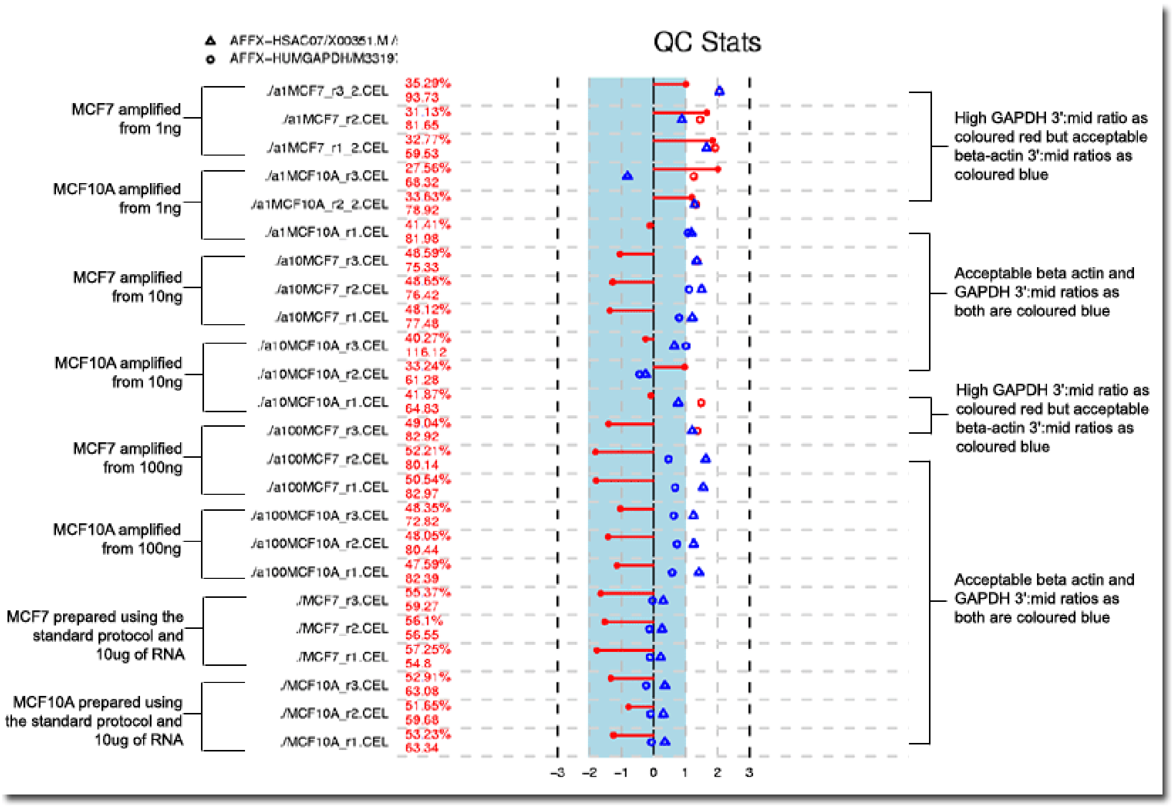
3. Number of genes called present

Although the number of genes called present (% present calls) shows a broad spread in values across the whole experiment (27-57%) there is good general agreement between samples in each replicate group and between each experimental condition. The qc plot has coloured these numbers red to indicate that there is a spread of more than 10% across the whole experiment, but this needs to be considered in conjunction with the other qc metrics. If we consider percent present and scale factor together, it can be seen that the 1ng samples that have lower % present calls also have higher scale factors (all 1ng MCF7 samples and replicates 2,3 of the MCF10 set These factors certainly warrant further attention. Higher scale factors in conjunction with lower percent present calls may indicate that less RNA has been hybridised to these chips.

4. 3' to 5' ratios for β-actin and GAPDH
At first glance the results for the 3':5' ratios in the above figure appear to be less than satisfactory, with many of the GAPDH and β-actin ratios being flagged.. This may indicate that theses samples contain degraded RNA. However these also coincide with samples prepared using the amplification protocol, and Affymetrix suggest that in these situations it is often more appropriate to assess the 3':mid ratios. All samples prepared using the standard protocol have β-actin and GAPDH 3':5' ratios that fall within the recommended values.

If 3':mid ratios are plotted (using the command `plot(qc.data,usemid=T)`; see figure below) then all samples have acceptable β-actin 3':mid ratios. However, some of the samples still have high GAPDH ratios.

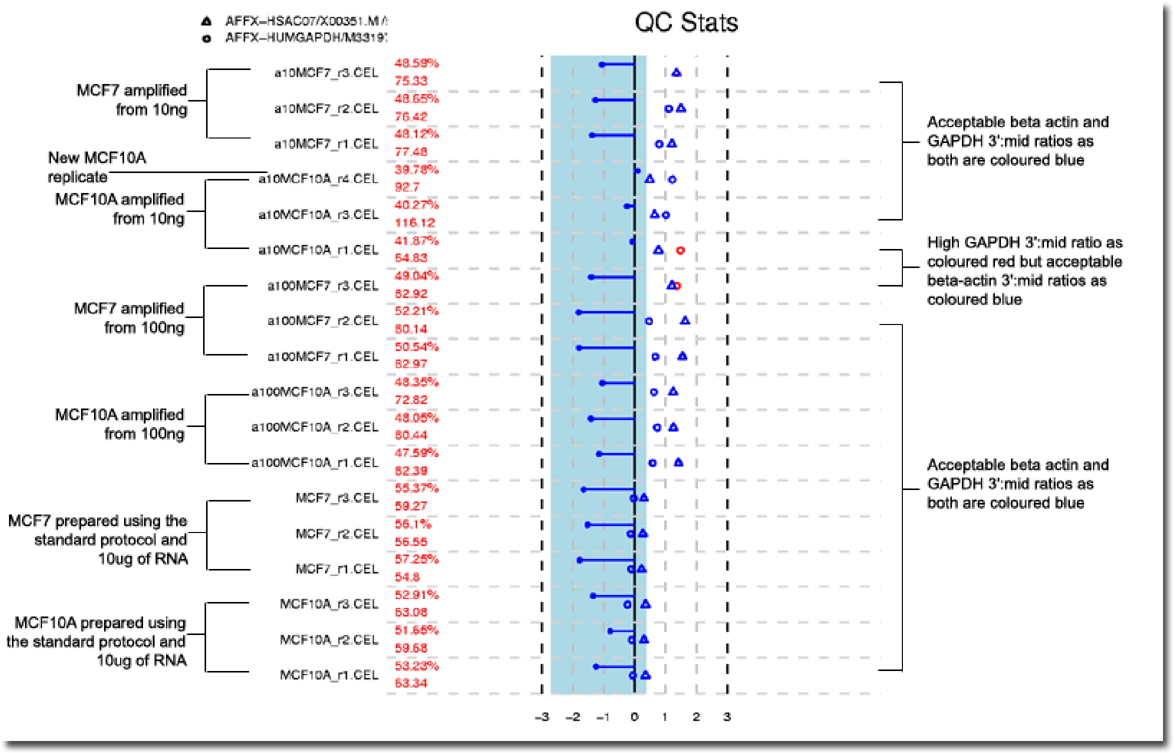
```
# R code used to generate the plot
> plot(qc.data, usemid=T)
```



QC conclusions

For both cell lines, all 1ng samples were discounted from any further analysis on the basis of these QC data. This was because the majority of the replicates in this group (all the MCF7 replicates and MCF10A replicates 2 and 3) had higher scale factors than the any of the other groups of samples, lower % present calls and higher GAPDH 3':mid ratios. These observations suggested that there could be degraded RNA present within the, samples or that the amplification protocol had not worked successfully for for the 1ng samples, resulting in less RNA being hybridised to the chip. Furthermore because of its lower percent present call and higher scale factor, the 2nd replicate in MCF10A amplified from 10ng was repeated, and the original chip discarded from any further analysis. The QC plot for the new data set is shown below and uses 3':mid ratios for both β-actin and GAPDH.

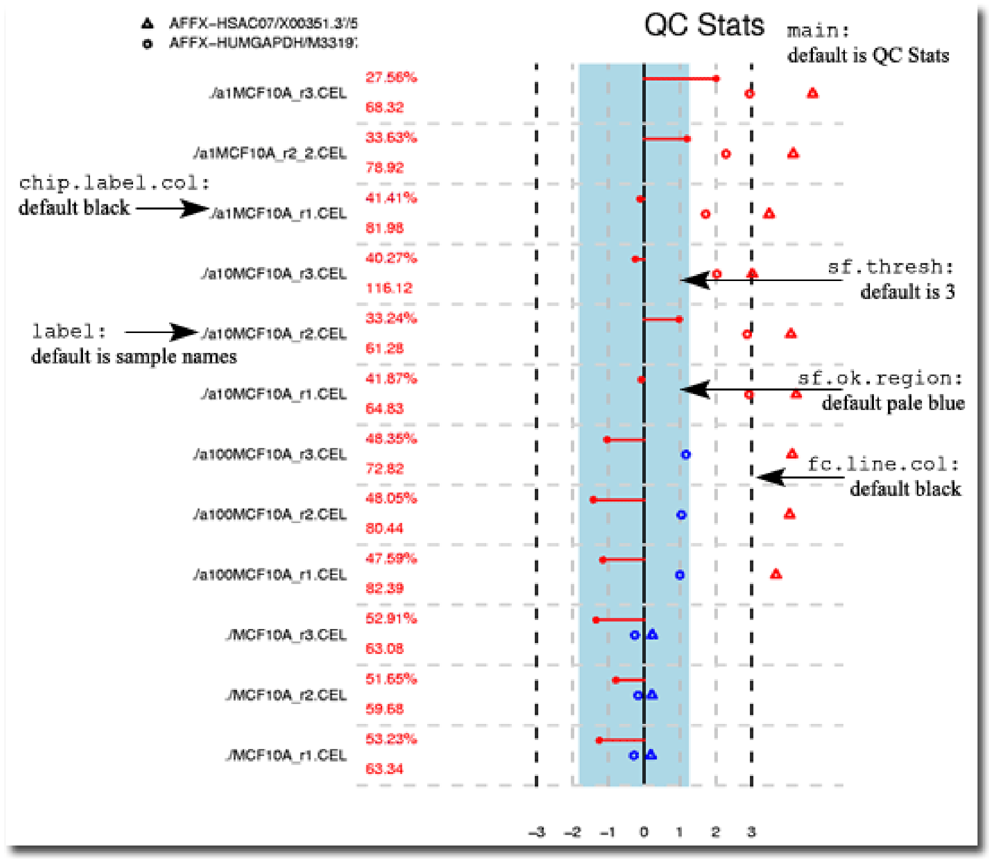
```
# R code used to generate the plot
> new.eset <- call.exprs(new.ampli, "mas5")
> qc.new <- qc(new.ampli, new.eset)
> plot(qc.new,usemid=T)
```



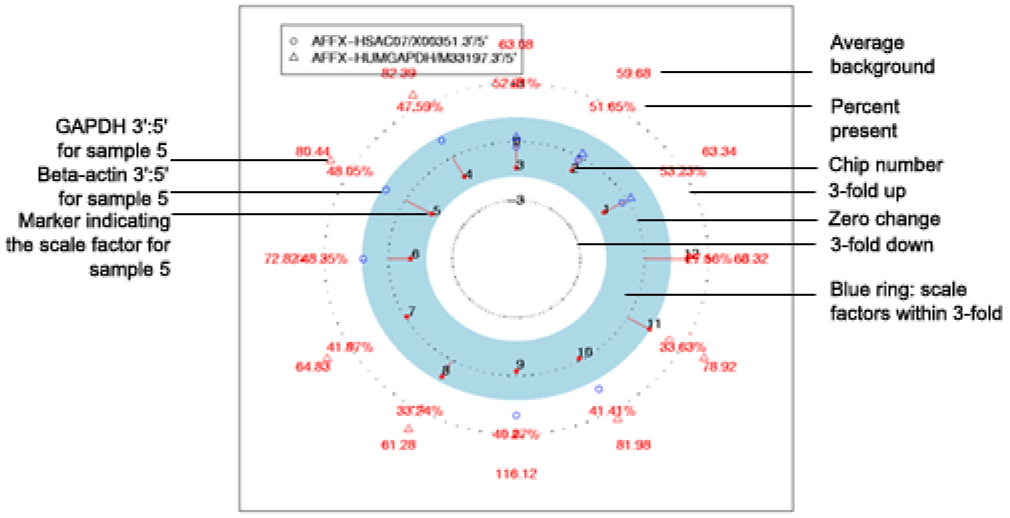
Although there is still a spread in the average background, because in general, the remaining QC metrics fall within the current guidelines (in particular the scale factors for all chips are within 3-fold of one another) this was not considered significant enough to reject arrays. Similarly, two samples (replicate 1 MCF10A amplified from 10ng and replicate 3 MCF7 amplified from 100ng) have higher GAPDH ratios, however, this threshold is fairly stringent, and in the context of the other QC metrics, it was decided not to fail these arrays.

Customising the QC plot

Although the qc plot is generated using a call to `plot()`, what actually happens is that the `QCStats` object is passed to the `plot.qc.stats` function that does the work. This function can take a number of parameters, detailed below.



fc.line.col: the colour to mark the dotted horizontal dotted lines with, default is black
sf.ok.region: the colour to mark the region in which scale factors lie within appropriate bounds, default is pale blue
chip.label.col: the colour to mark the fold change lines with, default is black
sf.thresh: Scale factors must be within this threshold of the mean scale factor for all chips, default is 3. If any scale factors fall outside this region, then the line connecting them to the horizontal dotted 0 line is coloured red if they all fall within 3-fold of the mean scale factor then this line is coloured blue.
gdh.thresh: GAPDH ratios (either 3':5' or 3':mid, depending on whether or not usemid is set to true or not) must be within this threshold of the mean scale factor for all chips, default is 1.25. If they fall within this limit then they are plotted in blue if they fall outside this limit they are plotted in red.
ba.thresh: β-actin ratios (either 3':5' or 3':mid, depending on whether or not usemid is set to true or not) must be within this threshold of the mean scale factor for all chips, default is 3. If they fall within this limit then they are plotted in blue if they fall outside this limit they are plotted in red.
present.thresh: The percentage of genes called present must lie within this range, the default is 10%.
bg.thresh: Array backgrounds must lie within this range, the default is 20 units.
label: A vector containing alternative sample names.
main: A title for the plot, the default is "QC Stats".
usemid: If true then the 3'Mid ratios of β-actin and GAPDH are plotted otherwise the 3':5' ratios are plotted.
type: If 'l' then the plot is constructed as detailed above and it is displayed vertically, if 'c' then the plot is displayed as a circle and chips are numbered sequentially from 1.


AFFX-HSAC07/X00351 3'/5'
AFFX-HUMGAPDH/M33197 3'/5'
Average background
Percent present
Chip number
3-fold up
Zero change
3-fold down
Blue ring: scale factors within 3-fold
GAPDH 3':5' for sample 5
Beta-actin 3':5' for sample 5
Marker indicating the scale factor for sample 5