Chapter 1

DNA MicroArray Data Analysis

The analysis of microarray data to produce lists of differentially expressed genes has several steps which can differ based on the type of data being assayed. However, all data follows the same general pipeline which involves reading raw data, quality assessing the data, removing bad spots/arrays from further analysis, preprocessing the data and calculating differential expression by statistical analysis. This list of differentially expressed genes can subsequently be annotated with useful information that explains the various genes’ function, for example, gene ontology. I will now explain in more detail how this data analysis pipeline is followed for the types of data supported by this system.

1.1 Preprocessing of Microarray Data

Before any kind of microarray data can be analysed for differential expression several steps must be taken. Raw data must be quality assessed to ensure its integrity. Unprocessed raw data will always be subject to some form of technical variation and thus must be preprocessed to remove as many unwanted sources of variation as is possible, to ensure that results are of the highest attainable level of accuracy. Ideally, the data being assayed should be preprocessed using several different methods, the results of which should be compared to identify which method is of the highest level of suitability. The most appropriate method should then be used to preprocess the raw data before differential expression analysis.

1.1.1 Preprocessing Affymetrix GeneChip Arrays

Because of the design of these kinds of chips, the steps that need to be taken before differential expression analysis are slightly more elaborate than for cDNA arrays, which we will outline later in the chapter.
Background Correction

The first step is generally to background correct the intensity reading for each spot. Background fluorescence can arise from many sources, such as non-specific binding of labelled sample to the array surface, processing effects such as deposits left after the wash stage or optical noise from the scanner. There is always some level of background noise, even if nothing but sterile water is labeled and hybridised to the array, some fluorescence will still be picked up by the scanner. Different algorithms will use different methods of background correction. The popular Robust Multi-Array Analysis (RMA) algorithm, for example, uses the convolution of signal and noise distributions.

Normalization

The next stage is normalization. The purpose of this step is to adjust data for technical variation, as opposed to biological differences between the samples. There will always be slight discrepancy between the hybridisation processes for each array and these variations tend to lead to scaling differences between the overall fluorescence intensity levels of various arrays. For example the quantity of RNA in a sample, the amount of time for which a sample spends hybridising or the volume of a sample can all introduce significant variance. Even subtle physical differences between arrays or between the scanners used to read arrays can have an effect.

Put simply, normalization ensures that when comparing expression levels of different arrays, that we are, as much as is possible, comparing like with like. Studies have shown that the normalization method used has a significant difference on final differential expression levels, so it is vital to choose an appropriate method.

PM Correction

As stated previously, PM probes on the GeneChip measure both the relative abundance of the corresponding gene and the amount of non-specific binding, which arises when mRNA binds to a probe which is not targeting it. MM probes are designed to give a measure for non-specific binding of their corresponding PM probe. It then seems obvious that the MM values should be subtracted from their corresponding PM values as a first step in the analysis process.

In reality however, this does not work, because generally about 30% of MMs are actually larger than their corresponding PMs. This is because, as well as measuring background signal, high volumes of mRNA targeted intentionally by the PM probes tend to also bind to MMs. Many of the most popular preprocessing
methods solve this problem by simply ignoring the MM probes altogether and PM values are corrected for non-specific binding using other methods.

Summarisation

We have already seen how GeneChip arrays work by using 11 different PM spots to target 11 separate 25 base long sections of a target genes mRNA. The final step in preprocessing GeneChip Data is to summarise the data from these 11 separate probes into an expression value for the gene in question. There are a number of different ways that this can be achieved, but the end result is always a single expression value for each gene on each chip.

1.1.2 Preprocessing Methods Implemented for Affymetrix GeneChip Array

Having introduced the general pipeline followed to preprocess Affymetrix microarray data, we will outline some of the preprocessing methods implemented by this system and describe their operation as well as justifying their inclusion.

There are a number of popular composite preprocessing algorithms. These algorithms implement the four preprocessing steps outlined above and output background corrected and normalized expression measures for each gene on each array. The preproessing methods implemented by this system are as follows.

MicroArray Suite 5.0 (MAS5)

MAS5 is an algorithm developed by Affymetrix and is described in their white paper “Statistical Algorithms Description Document” (2002) [?]. This algorithm background corrects both PM and MM probes; MMs are then converted into ideal mismatches, where their values are always smaller than their corresponding PM values. Remember than approximately 30% of the time MM values are greater than their PMs. If MM < PM, then MM value is left unchanged. A robust mean over the $\log_2$ transformed differences between PMs and the already calculated ideal mismatch is computed. Expression values are normalized by setting the trimmed mean of the original signals of each chip to a prespecified value. Hence, MAS5 data is normalized after summarisation, not before, as in many other algorithms.

Probe Logarithmic Intensity Error Estimation (PLIER)

PLIER is the current recommended algorithm from Affymetrix. Affymetrix claim that the algorithm improves on MAS5 by introducing a higher reproducibility of signal (lower coefficient of variation) without loss of accuracy; higher sensitivity to changes in abundance for targets near background and dynamic weighting of
the most informative probes in a dataset to determine signal [?]. In this system the PLIER algorithm is modified to include quantile normalization as PLIER does not normalize data by default.

**Robust Multi-Array Analysis (RMA)**

RMA is an academic alternative to Affymetrix’s algorithms for converting probe level data to gene expression measures. This method is distinct from Affymetrix’s methods in that it completely ignores the MM probe readings; the inventors of the algorithm claim that the MM probes introduce more noise and that, while acknowledging that these probes do provide useful information, have not, at the time of publication of the method, found a productive way to use it [?].

The methods work by adjusting for background noise on a raw intensity scale, which does not lead to negative background corrected values. The $\log_2$ transformed value of each background corrected PM probe is obtained and these values are normalized using quantiles normalization, which was developed by Bolstad et al. (2003) [?]. Robust multi-array analysis is then carried out on the quantiles [?].

**GeneChip RMA (GCRMA)**

GCRMA is largely based on RMA and in fact only differs in the background correction step where it uses probe sequence information to help estimate the background. This leads to improved accuracy in fold changes but at the expense of marginally lower precision [?].

**Other Methods Implemented**

The system can also carry out a preprocessing method by which the user can manually create the algorithm used, by specifying explicitly which of a selection of available functions, should be applied at each of the various stages, the options available to the user are as follows.

- **Background Correction:**
  - Mas5
  - RMA
  - RMA2

- **Normalization:**
  - Constant
The above options can be combined as the user desires to tailor preprocessing to their needs. This route is not recommended for novice users.

1.1.3 Preprocessing of cDNA Data

The general steps followed when preprocessing cDNA data are quite similar to the above. The main differences are that there is no need for PM correction, as there are no MM probes on cDNA arrays and that there is no summarisation stage, as each gene is represented by a single probe.

Background Correction

Background fluorescence occurs virtually identically in cDNA arrays as it does as previously described in oligonucleotide arrays [?]. The methods used to correct for background noise are described below.
Normalizing Within Arrays

There are a number of reasons that this step is performed for cDNA arrays. As noted by Smyth (2003) imbalances between the red (Cy5) and green (Cy3) dyes of cDNA arrays may arise from differences between the labelling efficiencies or scanning properties of the two dyes, complicated perhaps by, for example, the use of different scanners or different settings.

If the imbalance is more complicated than a simple scaling of one channel relative to the other, as it usually is, then the dye bias is a function of intensity and normalization will need to be intensity dependent. The dye-bias will also generally vary with spatial position on the slide. Positions on a slide may differ because of differences between the print-tips on the array printer, variation over the course of the print-run, non-uniformity in the hybridisation, or from artifacts on the surface of the array which affect one colour more than the other. [?]

Normalizing Between Arrays

Similarly to as outlined for oligonucleotide microarrays, cDNA arrays often suffer substantial scale differences because of technical variation, which could be down to any number of factors. Performing normalization between arrays will compensate for such effects and thus yield more reliable results.

1.1.4 Preprocessing Methods Implemented for cDNA Arrays

There are a large number of methods available for preprocessing of dual dye data. The system implements the following methods.

- Background Correction [?]:
  - Subtract
  - Half
  - Minimum
  - MovingMin
  - Edwards
  - NormExp
  - RMA

- Normalize Between Arrays [?]:
  - Aquantile
1.1.5 Preprocessing of Single Dye Arrays

The VSN method \cite{?} has been implemented to handle preprocessing of single channel data, such as that of Exiqon miRNA arrays. The function calibrates for sample-to-sample variations through shifting and scaling, and transforms the intensities to a scale where the variance is independent of the mean intensity. It combines background correction and normalization into one single procedure. For a matrix $x_{ki}$, with $k$ counting the probes and $i$ the arrays, the function fits a normalization transformation

$$x_{ki} \rightarrow h_i(x_{ki}) = \text{glog} \left( \frac{x_{ki} - a_i}{b_i} \right)$$  \hspace{1cm} (1.1)$$

where $b_i$ is the scale parameter for array $i$, $a_i$ is a background offset and glog is the generalised logarithm as described by Rocke and Durbin (2003) \cite{?}.

1.2 Data Quality Assessment Methods Implemented in System

Having introduced preprocessing of both Affymetrix GeneChip and cDNA microarray data, we now introduce and illustrate the importance of, the concept of quality assessment of microarrays data.

Quality assessment is an important phase that applies to analysis of all types of microarrays. Quality assessment of data ensures that the best use is made of the
information available and ensures meaningful results at the end of an analysis. It also aids us in choosing an appropriate preprocessing method, as data can be examined and visualised before and after preprocessing, where the impact of various algorithms can be compared and contrasted; a large number of tools have been implemented to see what effect the steps taken in preprocessing have had on the raw data.

These tools include visualisation plots as well as specific metrics that can be examined to assess whether discrepancies can be corrected by preprocessing, or that an array should be excluded in further analysis, or if necessary redone in the laboratory.

1.2.1 Quality Assessment of Affymetrix Genechip Data

There are a number of useful tools implemented to assess the quality of GeneChip data. We will now proceed to outline them and their various uses, using an example dataset.

The dataset being used to demonstrate preprocessing and quality assessment of GeneChip microarray data is from an experiment to determine the effects of negative energy balance on the postpartum cow. The bovine version of the Affymetrix GeneChip was used in this experiment. A set of six arrays from a negative energy balance group are compared to a set of six control arrays in order to determine differential gene expression.
Boxplot

A boxplot is a convenient means by which to compare the probe intensity levels between the arrays of a dataset. Either end of the box represents the upper and lower quartile. The line in the middle of the box represents the median. Horizontal lines, connected to the box by “whiskers”, indicate the largest and smallest values not considered outliers. Outliers are values that lie more than 1.5 times the interquartile range from the first of third quartile (the edges of the box); they are represented by a small circle.

If one or more arrays have intensity levels which are drastically different from the rest of the arrays, this may indicate a problem with these arrays. These kinds of problems can however sometimes be corrected by normalization. For microarray data, these graphs are always constructed using $\log_2$ transformed probe intensity values, as the graph would be virtually unreadable using raw values, as you can see below, where raw values are juxtaposed with $\log_2$ transformed values.

The boxplot of log transformed intensity levels in the above example communicates some useful information. As can be seen the fourth array from the left has noticeably higher overall probe intensity readings than any of the other arrays. This could be an early indication of a problem with this array. We need to perform further investigation and establish if this discrepancy an be corrected by normalization. The Figures on the next page show boxplots of probe intensity levels following, RMA, GCRMA, MAS5 and qPLIER preprocessing algorithms.
The above plots give an interesting picture of how different algorithms affect the raw data in significantly different ways. We have a good indication that normalization could solve the scaling problem of our rogue array. We will however need more much more information in order to make an informed decision as to whether this array should or shouldn’t be included in analysis and which preprocessing method should be selected.
Histogram

A histogram of array intensity levels tells us quite a similar story to that of a boxplot. It is again used to visualise the spread of data and compare and contrast probe intensity between the arrays of the dataset. The x-axis represents probe density level and the y-axis indicates probe intensity. This plot provides us with a slightly more detailed picture and there are a number of inferences that can be made from these plots; a bimodal distribution in the raw data, for example, is often indicative of an array containing a spacial artifact and an array which is shifted to the right often has abnormally high background interference.

As you can see from the plot of our raw data below, the array “NS7.CEL” is once again a problem, being shifted slightly to the right, which as stated above could indicate high levels of background noise. This point is worth continued investigation.

For comparison purposes I have also included an image of RMA preprocessed values. Even with normalization the same array still has the highest overall values.

Figure 1.7: Raw probe intensity values plotted on a histogram

Figure 1.8: RMA Preprocessed Histogram
Principal Componenet Analysis

Principal Componenet Analysis (PCA) is used to reduce multidimensional datasets to lower dimensions for analysis; it is a technique that can determine the key features of high-dimensional datasets. In the context of microarray analysis, PCA essentially clusters arrays by groups of the most significantly dysregulated probes. Clustering first by the most significant group, then by progressively less significant groups.

Given the experimental design of the dataset that we are attempting to analyse here, where microarrays belong to only two distinct groups, a control group and a treatment group, there should be a clear separation of both groups of arrays by the principle component, because, assuming experimental conditions were properly controlled, most of the variance in expression level should have been introduced because of the conditions under scrutiny, in our case, negative energy balance.

The following figures show PCA plots of the unprocessed, RMA preprocessed and MAS5 preprocessed data intensity levels. These plots provide further evidence that rogue array “NS7.CEL” contains high levels of background noise or otherwise compromised data, which cannot be corrected by normalization, as becomes clear from its non-clustering with its fellow group members before or after preprocessing. Given this and previous information we can be quite confident that this array shouldn’t be included in differential expression analysis.

The key to identify individual arrays on these plots is at the end of the next page.

![PCA plot of raw data. Note non-grouping of NS7.CEL.](image)
Figure 1.10: PCA plot of RMA preprocessed data. Note improved overall clustering but continued non-grouping of NS7.CEL.

Figure 1.11: PCA plot of MAS5 preprocessed data. Once again the situation is not ameliorated for NS7.CEL.
RNA Degradation Plot

Another quality assessment tool that has been implemented is the RNA degradation plot, which gives a good indication of the quality of the sample that has been hybridised to the array. mRNA degradation occurs when the molecule begins to break down and is therefore ineffective in determining gene expression. Because this kind of degradation starts at the 5’ end of the molecule and progresses to the 3’ end it can be easily measured using oligonucleotide arrays, where each PM probe is numbered sequentially from the 5’ end of the targeted mRNA transcript.

When RNA degradation is advanced, PM probe intensity at the 3’ end of a probeset should be elevated when compared with the 5’ end.

When dealing with high quality RNA a slope of between .5 and 1.7 is typical, depending on the type of array; slopes that exceed these values by a factor of 2 or higher could indicate excessive degradation, the actual value is however less important than agreement between the chips, because if all the arrays have similar slopes then comparisons within genes across the arrays may still be valid [?].

Shown below is an RNA degradation plot for the dataset which we are assaying. The slope falls within the recommended range, which indicates that all of the samples were of good quality. There is a very strong correlation between the various arrays in the dataset.

Figure 1.12: RNA Degradation Plot
Simple Affy Plot and Affymetrix Recommended Metrics

Affymetrix recommends the examination of a number of quantities for quality assessment of GeneChip data; these metrics have been included in the quality assessment tools of this system. These are specifically, averages background, scale factor and percent present calls.

Average background indicates the level of background noise a chip is experiencing. There are several reasons that chips may have significantly different average background intensities. It might be simply that the overall signal from the array is greater, because different amounts of RNA were present during hybridisation, or that hybridisation was more efficient, thus producing a more fluorescent chip. It is recommended that these values should be similar across all chips [?].

Scaling factor refers to the level of scaling applied to an array when normalized using Affymetrix’s MAS5 algorithm. By default MAS5 scales the intensity of each array so that they all have the same mean. So scaling factor is a measure of how far a chip’s overall values are scaled because of this. Affymetrix recommends that scale factors be within 3-fold of each other [?].

Percent Present calls are generated by looking at the difference between PM and MM probes for each pair in a probeset and simply represents the percentage of probesets called present on an array. Probesets are flagged marginal or absent when the PM values for that probeset are not considered to be significantly above the MM probes. Similarly to scale factors, significant variations in percent present calls across the arrays in a study should be treated with caution [?]. Again it is recommended by Affymetrix that these values be similar.

Below are tables of these values for our example dataset.

<table>
<thead>
<tr>
<th>Table 1.1: Background Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>N12S.CEL</td>
</tr>
<tr>
<td>67.49</td>
</tr>
<tr>
<td>N10S.CEL</td>
</tr>
<tr>
<td>65.61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 1.2: Scale Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>N12S.CEL</td>
</tr>
<tr>
<td>0.44</td>
</tr>
<tr>
<td>N10S.CEL</td>
</tr>
<tr>
<td>0.40</td>
</tr>
</tbody>
</table>
Table 1.3: Percent Present Calls

<table>
<thead>
<tr>
<th>Sample</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>N12S.CEL</td>
<td>57.73%</td>
</tr>
<tr>
<td>N11S.CEL</td>
<td>58.61%</td>
</tr>
<tr>
<td>N8S.CEL</td>
<td>57.69%</td>
</tr>
<tr>
<td>P1S.CEL</td>
<td>57.89%</td>
</tr>
<tr>
<td>P4S.CEL</td>
<td>57.75%</td>
</tr>
<tr>
<td>P5S.CEL</td>
<td>58.16%</td>
</tr>
<tr>
<td>N10S.CEL</td>
<td>58.97%</td>
</tr>
<tr>
<td>P3S.CEL</td>
<td>59.57%</td>
</tr>
<tr>
<td>P6S.CEL</td>
<td>58.17%</td>
</tr>
<tr>
<td>P2S.CEL</td>
<td>59.46%</td>
</tr>
<tr>
<td>N7S.CEL</td>
<td>54.78%</td>
</tr>
<tr>
<td>N9S.CEL</td>
<td>58.90%</td>
</tr>
</tbody>
</table>

None of the values in the tables above are alone particular cause for concern. It is worth noting however that the array we previously identified as an outlier “NS7.CEL” does contain the highest level of background noise, which we previously suspected could have been part of the cause of its problems.

Shown below is a simpleAffy plot, which is visual representation of some of the data above. The plot is labeled with some explanations of how to read it.

Figure 1.13: SimpleAffy Plot
Probe Level Models and Pseudo Array Images

The system implements functions that fit the following linear model to probe level data using robust regression procedures described by Huber (1981) [?] and implemented in R by the rlm() function from the package MASS by Venables and Ripley (2002) [?]. This will be further discussed in chapter 3.

\[ \log(Y_{gij}) = \theta_{gi} + \phi_{gj} + \varepsilon_{gij} \] (1.2)

The above equation is referred to as a probe level model; \( \theta_{gi} \) represents the log transformed expression level for gene \( g \) on array \( i \), \( \phi_{gj} \) is the effect of the \( j \)-th probe representing gene \( i \), and \( \varepsilon_{gij} \) is the error measurement for the probe.

The system can be used to fit the above model; one of the main benefits of which is that numerous useful quality assessment tools can be derived from the output of the PLM fitting procedure [?].

Below we show four pseudo images of an array from one of Bioconductor’s prepackaged sample datasets. This particular example to illustrates how PLM can be used to identify artifacts on an array. The individual images, reading left to right, top to bottom are, raw probe intensities, weights used by robust regression to downweight outliers, residuals and signed residuals.

It is clear from the images that the array contains an artifact, which manifests itself as a visible ring in all three PLM images, but, due to the strong probe effect \( \phi \), is not obvious in the image of probe intensities.

Figure 1.14: Array Images
Relative Log Expression and NUSE Plots

These are two further plots which can be constructed based on the probe level model that we have fitted above.

The Relative Log Expression (RLE) plot shows, for each array, the deviation of gene expression level from the median gene expression level for that gene across all arrays. An array with quality problems may show significantly different values than the majority of arrays, resulting in an RLE box with greater spread or a median which deviates from 0. To construct this plot, the log estimates of expression $\theta_{gi}$ for each gene $g$ on each array $i$ are computed. The median value across all arrays for each gene $m_g$ is computed and relative log expression is defined as $M_{gi} = \theta_{gi} - m_g$. An array with quality problems may result in a box that has greater spread and/or is not centred on $M = 0$.

The Normalized Unscaled Standard Error (NUSE) plot portrays the chip-wise distribution of standard error estimates, obtained for each gene on each array. To account for the fact that variability differs considerably between each genes, the error estimates are standardised so that the median standard error across arrays is 1 for each gene. The NUSE values are calculated as follows:[1]

\[
\text{NUSE}(\theta_{gi}) = \frac{\text{SE}(\theta_{gi})}{\text{med}_i(\text{SE}(\theta_{gi}))} \tag{1.3}
\]

NUSE and RLE plots of our original bovine dataset are shown below. You can see that once again “NS7.CEL” is again slightly askew in both figures.
1.2.2 Quality Assessment of cDNA Data

There are also number of useful tools implemented to assess the quality of cDNA data. This subsection aims to outline these tools and their various uses.

The dataset which will be used to demonstrate preprocessing and quality assessment of cDNA microarray data is the Swirl dataset, which is one of the example datasets packaged with Bioconductor.

To give a very brief background; this experiment was carried out using zebrafish as a model organism to study the early development in vertebrates. Swirl is a point mutant in the BMP2 gene that affects the dorsal/ventral body axis. The main goal of the Swirl experiment is to identify genes with altered expression in the Swirl mutant compared to wild-type zebrafish. Each of the four arrays in the experiment compares RNA from the mutant Swirl zebrafish to that of the normal “wild-type” fish.

The following pages outline the cDNA quality assessed tools implemented in this system.
M-A Plots

$M$ and $A$ are two very commonly used variables in the analysis of dual dye arrays and understanding their meaning is a crucial concept in understanding this kind of analysis.

$A$ is defined by

$$A = \log_2 \sqrt{C_{y5} \cdot C_{y3}} = \frac{1}{2} \left[ \log_2(C_{y5}) + \log_2(C_{y3}) \right]$$  \hspace{1cm} (1.4)

$C_{y5}$ and $C_{y3}$ represent respectively the red and green dye intensities of a particular spot. So $A$ is the red and green intensities of a spot multiplied together, square rooted and log transformed. Thus it is essentially a measure of the total log transformed intensity of a spot. Essentially, if combined red and green intensities are high for a particular spot, then $A$ will also be high.

$M$ is defined as

$$M = \log_2 \frac{C_{y5}}{C_{y3}} = \log_2 \frac{C_{y5}}{C_{y3}} = \log_2(C_{y5}) - \log_2(C_{y3})$$  \hspace{1cm} (1.5)

So $M$ is the log transformed red dye intensity divided by the green dye intensity. It gives an indication of whether more of either the red or green dye binding to the array at a given spot.

The purpose of an MA-plot is to investigate intensity bias. If a disproportionate amount of spots on the plot are above or below the x-axis it could indicate a problem with an array. As before these kinds of problems can sometimes be addressed with normalization.

MA-plots can be viewed for a whole array, or for the individual print tip groups on an array. This diagram gives us a good indication of whether normalization within an array is needed.

Below are MA-plots of the first array, “swirl.1.spot”, in our example dataset. Plots are shown for both print-tip groups and the array as a whole.

Data for the normalized plots is created using the print-tip-group loess within array normalization technique, which is suitable for most kinds of data.
Figure 1.17: MA-plot of Raw Swirl.1.spot

Figure 1.18: MA-plot of print-tip-group loess normalized Swirl.1.spot

Figure 1.19: MA plots for raw print tip groups of Swirl.1.spot

Figure 1.20: MA plots of normalized print tip groups of raw Swirl.1.spot
Pseudo Array Images

As outlined previously, viewing array images can be a useful step in identifying artifacts on an array, that may lead to the arrays exclusion from an experiment. Included below are pseudo-images for another array in our experiment, this time “Swirl.2.spot”. Shown are foreground and background red and green images. The range of intensity values is also printed on the bottom of the image, this indicates what intensities the various colours represent. For example on our red foreground image, the intensity range is 7.5 to 15.6, indicating that a spot with log transformed intensity level of 7.6 is represented by pure white and a spot with intensity of 15.6 is represented by pure red, while values in between are represented by colours varying progressively from white to red.

Also shown are images of M, the log ratio of red and green intensities, for both raw and print-tip-group loess normalized data.

Figure 1.21: Array image of green foreground intensities

Figure 1.22: Array image of green background intensities
Figure 1.23: Array image of red foreground intensities

Figure 1.24: Array image of red background intensities

Figure 1.25: Array images of M (log-ratios) of raw data

Figure 1.26: Array images of M (log-ratios) of normalized data

**Boxplot of M (Log Ratios) and Intensity Histogram**

This plot is useful in assessing whether normalization between arrays should be performed. This kind of normalization should be performed if there are scaling differences between the different arrays. As can be seen from the first boxplot of raw M values (Fig. 1.27), there are significant scaling differences, which means that in this case between array normalization should be performed on the Swirl dataset. The second boxplot shows the arrays normalized using quantile normalization.

It would appear from the boxplots that the scaling differences have been solved using normalization.
The kind of intensity histogram below follows the exact same principal as the histogram already described for oligonucleotide arrays, save for the fact that in this case each array is represented by both a red and a green channel.

An intensity histogram is shown for both raw and quantiles normalized data. Note that boxplots of red and green foreground and background intensity levels can also be viewed.

Figure 1.27: Boxplot of M for raw data

Figure 1.28: Boxplot of M for quantile normalized data

Figure 1.29: Raw Intensities Plot

Figure 1.30: Quantile Normalized Intensities Plot
1.2.3 Quality Assessment of Single Dye Data

Support for single channel platforms like Exiqon miRNA arrays in Bioconductor is still in something of an experimental stage and can be somewhat *ad-hoc*; as already stated, only the VSN preprocessing method is available. Quality control options are slightly more limited than for other platforms, but there is still enough available for a user to make a reasonable judgement about the integrity of a dataset’s constituent arrays.

The system implements many of the same plots as before. Available for assessment are, array images of both foreground and background intensities, boxplots of raw and preprocessed foreground and background intensities, density plots of raw and preprocessed data and PCA and accompanying scree plots of raw and preprocessed data. All of these plots should be assayed in a similar manner as already described for other platforms.

1.3 Calculating Differential Expression

Differential expression analysis of microarray data is fraught with many classical statistical issues, such as appropriate test statistics, replicate structure, sample size, outlier detection and statistical significance of results. The original and simplest approach to identifying differentially expressed genes was to use a fold change criteria; selecting cutoff was something of an *ad-hoc* procedure; a 2-fold change was however thought as being a suitable cutoff. This selection process is however, completely biased towards individual genes with large fold changes and completely disregards the fact that groups of related genes showing smaller deviations could be just as important and also does not allow for assessment of significance of expression differences in the presence of biological and experimental variation [?].

There are a number of statistical tests available that can be applied to assess differential expression between populations of microarray data, such as the *t*-test, which can be used to assess the statistical probability that, given the number of samples available, the true expression levels for a given gene differ in the overall populations. In such an analysis, the number of samples is invariably far less than the number of genes which are being investigated. The number of genes could run into tens of thousands, but the number of arrays used will, due to overall cost or rarity of tissue samples, rarely exceed thirty, thus creating a multiple testing problem. For example, on an array of 25,000 genes, if even 5% are misinterpreted as being differentially expressed, or “false positives”, then \( \sim 1,250 \) genes will be construed as being differentially expressed when they are in fact not.

There are a number of solutions available to the problem of false positives which result from the large number of variables in a statistical test; they include
False Discovery Rate (FDR) developed by Benjamini and Hochberg (1995) \cite{Benjamini1995}, or the more stringent Bonferroni Method which controls the family-wise error rate. These and other methods can be applied to address the problem of false positives in microarray gene expression analysis.

The system developed during this project uses the functions available in Bioconductors LIMMA package to calculate differential expression of GeneChip, dual dye and single dye data, as the same principals can be applied to all of these data types.

Further to that, the system also implements the functionality of the more recent PUMA package, for analysis of GeneChip data.

Note that further technical details of how these packages are integrated will be discussed at a later point in this thesis.

1.3.1 The LIMMA Package

LIMMA is an R library which is part of the Bioconductor project and is used for the analysis of gene expression microarray data. It incorporates the use of linear models for assessment of differential expression. LIMMA provides the ability to analyse comparisons between many RNA targets simultaneously in complicated designed experiments. Empirical Bayesian methods are used to provide stable results even when the number of arrays is small.

The general procedure followed in analysis using the package is as follows.

This procedure first fits a linear model to the expression data for each probe. The expression data should be log-ratios $M$ for two-colour array platforms or log-expression values for one-channel platforms. The coefficients of the fitted models describe the differences between the RNA sources hybridised to the arrays, these coefficients are described by the design matrix. The probe-wise fitted model results are stored in a compact form suitable for further processing by other functions in the Limma package.

The fitted model object is then re-orientated from the coefficients of the original design matrix to any set of contrasts of the original coefficients. The coefficients, correlation matrix and unscaled standard deviations are then re-calculated in terms of the contrasts.

Finally, Empirical Bayes shrinkage is used to compute moderated t-statistics, moderated F-statistic, and B-statistic (log-odds of differential expression) by shrinkage of the standard errors towards a common value. This method has the advantage of being able to provide a stable result even when the number of arrays in an experiment is small \cite{Gilgenkrantz2004}.

Below are screen shots of the top ranked differentially expressed genes from the two datasets we introduced earlier. The GeneChip data (bovine dataset) was preprocessed using RMA; while the Swirl dataset, which is a dual dye experiment
where array image analysis was performed using Spotfire Software, was preprocessed using background subtraction, Print-tip-group Loess normalization within arrays and Quantile normalization between arrays. In both cases, the adjusted p-values are corrected for multiple testing using the Benjamini and Hochberg method.

These screen shots are of HTML tables output by the system. How these are created will become clear over the subsequent chapters.

<table>
<thead>
<tr>
<th>ID</th>
<th>Genes</th>
<th>LogFC</th>
<th>P.Value</th>
<th>adj.P.Val</th>
<th>AveExpr</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.22/18.1A1</td>
<td>7.29E+07</td>
<td>2.28E+07</td>
<td>0.03E+07</td>
<td>0.00E+00</td>
<td>8.29E+07</td>
</tr>
<tr>
<td>B.785.1S1</td>
<td>1.0E+08</td>
<td>5.2E+07</td>
<td>0.01E+00</td>
<td>0.00E+00</td>
<td>5.1E+07</td>
</tr>
<tr>
<td>B.1214.2S1</td>
<td>1.0E+08</td>
<td>6.0E+07</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>8.1E+07</td>
</tr>
<tr>
<td>B.5536.1S1</td>
<td>6.0E+07</td>
<td>3.1E+07</td>
<td>0.02E+00</td>
<td>0.01E+00</td>
<td>7.2E+07</td>
</tr>
<tr>
<td>B.1991.1S1</td>
<td>5.9E+07</td>
<td>3.6E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>6.0E+07</td>
</tr>
<tr>
<td>B.1398.4S1</td>
<td>1.2E+08</td>
<td>8.4E+07</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>6.9E+07</td>
</tr>
<tr>
<td>B.1214.3S1</td>
<td>9.0E+07</td>
<td>5.5E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>7.7E+07</td>
</tr>
<tr>
<td>B.5392.2S1</td>
<td>8.0E+07</td>
<td>8.1E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>6.4E+07</td>
</tr>
<tr>
<td>B.1966.8S1</td>
<td>7.9E+07</td>
<td>4.4E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>5.8E+07</td>
</tr>
<tr>
<td>B.5164.1S1</td>
<td>6.5E+07</td>
<td>3.7E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>5.7E+07</td>
</tr>
</tbody>
</table>

Figure 1.31: Top 10 genes for Bovine dataset

<table>
<thead>
<tr>
<th>ID</th>
<th>Genes</th>
<th>LogFC</th>
<th>P.Value</th>
<th>adj.P.Val</th>
<th>AveExpr</th>
</tr>
</thead>
<tbody>
<tr>
<td>ths41d</td>
<td>5.9E+07</td>
<td>4.3E+07</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>7.9E+07</td>
</tr>
<tr>
<td>control</td>
<td>1.9E+08</td>
<td>4.8E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>7.4E+07</td>
</tr>
<tr>
<td>control</td>
<td>2.1E+08</td>
<td>8.4E+07</td>
<td>0.00E+00</td>
<td>0.00E+00</td>
<td>6.8E+07</td>
</tr>
<tr>
<td>ths41d</td>
<td>1.5E+08</td>
<td>2.0E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>5.9E+07</td>
</tr>
<tr>
<td>ths22d</td>
<td>1.4E+08</td>
<td>2.5E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>3.9E+07</td>
</tr>
<tr>
<td>ths41d</td>
<td>5.9E+07</td>
<td>1.3E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>6.5E+07</td>
</tr>
<tr>
<td>ths41d</td>
<td>9.1E+07</td>
<td>3.3E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>5.4E+07</td>
</tr>
<tr>
<td>ths41d</td>
<td>5.9E+07</td>
<td>1.3E+07</td>
<td>0.01E+00</td>
<td>0.01E+00</td>
<td>6.5E+07</td>
</tr>
</tbody>
</table>

Figure 1.32: Top 10 genes for Swirl dataset

### 1.3.2 The PUMA Package

PUMA is an acronym for Propagating Uncertainty in Microarray Analysis [?]. It is a package that is specifically targeted at GeneChip data. Unlike previous analyses of Affymetrix GeneChip data, PUMA does not simply provide a point estimates of gene expression levels. The designers of PUMA argue that the original
set of \( \sim 11 \) probes contain much useful information about uncertainty associated with their final expression measure. Using probabilistic methods, it is possible to associate gene expression levels from probe level analysis with credibility intervals that quantify uncertainty associated with the estimate of target concentration in a sample. By propagating this uncertainty to downstream analyses, it is argued that the reliability of results is improved. Included in the package are summarisation, differential expression detection, clustering and PCA methods, together with useful plotting functions.

PUMA uses the multi-mgMOS preprocessing method\([7]\), which uses Bayesian methods to associate credibility intervals with expression levels.

For PCA, a noise-propagation in principal components analysis method\([8]\) is used, which propagates the expression level uncertainty to improve the results of PCA.

By default, genes are ranked for differential expression using the Probability of Positive Log Ratio (PPLR) method\([9]\) which combines uncertainty information from replicated experiments in order to obtain point estimates and standard errors of the expression levels within each condition. These point estimates and standard errors can then be used to obtain a PPLR score for each probeset, which can then be used to rank probesets by probability of differential expression between two conditions \([10]\).

1.4 Use of Remapped Probe Sets For GeneChip Arrays

As already outlined, most GeneChip arrays use 11 different 25-base long probes to target specific genes.

A problem is however introduced by the ever changing nature of knowledge of genomic sequences of different organisms. As such knowledge evolves, it has become clear that the original probe to transcript mappings assigned in an array’s Chip Definition File (CDF), defined initially by the manufacturer, are in certain cases, known to be no longer entirely accurate. In simple terms, some probes are not targeting the sequence that they were originally thought to be targeting.

Because of this a number of groups have developed alternative probe to probeset mappings, which are defined in remapped Chip Definition Files.

This system gives the user the option of using some of the remapped CDF packages created by the AffyProbeMiner project \([11]\), as an alternative to the default Affymetrix CDFs. AffyProbeMiner regroups probes in the GeneChip into new probesets according to the verified complete coding sequences available in GeneBank and RefSeq databases. This remapping has been shown to affect 20-30% of all probesets, with genes shown to be differentially expressed using the
default CDF file showing only a 50% overlap with an analysis based on the new CDF, but the remapped probesets are more consistent with the latest genomic sequencing information and therefore provide a more reliable measure of a gene's true expression level [?].

Below is a list of the top ten genes in our bovine dataset calculated using Affyprobeminers remapped CDF file and the methods in the PUMA package. Note that the ID column contains EntrezGene gene IDs, as opposed to standard Affymetrix identifiers, which are assigned by default by Affyprobeminer.

<table>
<thead>
<tr>
<th>#</th>
<th>ID</th>
<th>Fold_Change</th>
<th>PPLR</th>
<th>p_Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>201052</td>
<td>-0.63562295620659</td>
<td>0.393295956089e-13</td>
<td>1.87660997852e-12</td>
</tr>
<tr>
<td>2</td>
<td>532851</td>
<td>0.420060577</td>
<td>0.99999999983</td>
<td>3.8815642205e-11</td>
</tr>
<tr>
<td>3</td>
<td>282188</td>
<td>1.25679201128</td>
<td>0.999999990259</td>
<td>1.4845395318e-09</td>
</tr>
<tr>
<td>4</td>
<td>282751</td>
<td>0.423860051404</td>
<td>0.999999995402</td>
<td>9.15573726275e-09</td>
</tr>
<tr>
<td>5</td>
<td>615436</td>
<td>0.695549652064</td>
<td>0.999999991034</td>
<td>1.7931283125e-08</td>
</tr>
<tr>
<td>6</td>
<td>614262</td>
<td>-0.511747433069</td>
<td>2.60624623372e-06</td>
<td>5.2124925265e-08</td>
</tr>
<tr>
<td>7</td>
<td>515903</td>
<td>0.779850344029</td>
<td>0.9999999952283</td>
<td>9.54343923887e-08</td>
</tr>
<tr>
<td>8</td>
<td>201574</td>
<td>0.307087117749</td>
<td>0.999999981452</td>
<td>2.3706281275e-07</td>
</tr>
<tr>
<td>9</td>
<td>100034674</td>
<td>0.334464729544</td>
<td>0.9999999815916</td>
<td>3.68167486053e-07</td>
</tr>
<tr>
<td>10</td>
<td>282977</td>
<td>0.783694528379</td>
<td>0.999999737808</td>
<td>5.2438443867e-07</td>
</tr>
</tbody>
</table>

Figure 1.33: Top 10 genes for Bovine dataset (PUMA)