BioconductorBuntu Users Manual

Mr. Paul Geeleher, Dr. Dermot Morris, Dr. Aaron Golden and Professor John Hinde

This project attemts to build a user friendly interface to some of the DNA microarray packages included in the Bioconductor project.

Bioconductor is an existing open source software project that attempts to facilitate analysis of genomic data. It is a collection of packages for the statistical programming language R. Bioconductor is particularly useful in analysing microarray experiments. The problem is that the R programming language's command line interface is intimidating to many users who do not have a strong background in computing. This often leads to a situation where biologists will resort to using commercial software which often uses antiquated and much less effective statistical techniques, as well as being expensively priced. This project aims to bridge this gap by providing a user friendly web-based interface to the cutting edge statistical techniques of Bioconductor.

The analysis tools that we have constructed as part of this project facilitate straightforward analysis of microarray data in a web-based environment, addressing the most widely used microarray platforms and following a logical progression through an analysis pipeline that is both extensible and capable of addressing current needs.

The initial scope of this project primarily focused on analysis of Affymetrix GeneChip arrays. However the facilities for basic analysis of dual dye cDNA arrays and single dye Exiqon miRNA arrays have also now been included and provide a solid foundation for future development.

BioconductorBuntu is a custom distribution of Ubuntu Linux that wraps the analysis tools developed by the project in an easily installable and distributable format. The server is setup by running a very straight forward installation CD. The system is best installed on a dedicated server, allowing any individuals connected to the same network to make use of the analysis tools hosted there.

The first two chapters of this manual serve to provide an intoroduction to genomics and microarray analysis. These chapters can be skipped by advanced users.

Contents

Chapter 1

Introduction

1.1 Introduction to Genomics

1.1.1 What are Genes and DNA?

Eukaryotes are organisms whose cells are organized into complex structures enclosed within membranes. Most eukaryotic organisms, for example, human beings, contains billions of individual cells. Almost all of these cells contain, within each nucleus, the entire genome for that organism. This genome contains the organism's complete hereditary information in the form of deoxyribonucleic acid (DNA), that encodes a complete blueprint for all activities and structures within the organism.

In the human body, the genome consists of 23 pairs of chromosomes. One of each of this pair is inherited from the mother and the other from the father. Each chromosome is made of chains of DNA. DNA consists of two polymers made up of units called nucleotides. Each nucleotide consists of a deoxyribose sugar, a phosphate group and one of the four nitrogen bases, guanine, adenine, thymine and cytosine. These bases, which are usually represented by their first letters, G, A, T and C, are where hereditary genetic information is actually encoded. It is worth noting that one of the two strands of the DNA double helix will suffice to describe this information; this is because of complementary base pairing, whereby an A on one strand always binds to a T on the other and a C always binds to a G.

Genes are essentially segments of the DNA structure described above. Loosely speaking, a gene is a section of DNA that defines a single trait by encoding a particular pattern, about 27,000 of which exist in humans. Often though we are faced with a case where protein-coding sequences have no clear beginning or end; more technically, a gene is a locatable region of genomic sequence, corresponding to a unit of inheritance, which is associated with regulatory regions, also known as exons, transcribed regions, also known as introns and/or other functional sequence regions [18].

1.1.2 What are Proteins and How are They Created?

The main purpose of genes is to act as a blueprint in the creation of proteins. Proteins are made of amino acids and are responsible for the structure and activity of an organism at a cellular level. They are created as follows; starting at the 5' end (the leading end) of a gene and proceeding to the 3' end (the tail end), the information contained in the gene is transcribed into a messenger ribonucleic acid (mRNA) strand. This process is performed by an enzyme called RNA polymerase. After transcription this mRNA molecule leaves the nucleus of the cell where it is transcribed into a protein in a process called translation. This is performed by ribosomes, which read the code carried by mRNA molecules from the cell nucleus and create proteins combining any of the 20 amino acids in the body into complex polypeptide chains. These proteins are the building blocks of the organism. This process of translating a gene into a functional product is known as gene expression.

1.2 DNA Microarrays

DNA microarrays are a high throughput technology used to measure the expression levels of thousands of genes, in some cases all of the genes in a genome, simultaneously [12]. The fundamental idea behind most microarrays is to exploit complementary base pairing (see previous section) to measure the amount of the different types of mRNA molecules in a cell, thus indirectly measuring the expression levels of the genes that are responsible for the synthesis of those particular mRNA molecules.

The spots on a microarray contain single stranded DNA oligonucleotides called probes. Each of these spots will contain DNA which is of a complementary sequence to the specific mRNA molecule that corresponds to the gene that it is targeting. An mRNA molecule which is complementary to the probe in question, should hybridise to that probe, forming a strong mRNA-DNA bond. These mRNA molecules will have previously been labeled with fluorescent dye, which means that the amount of hybridisation that has taken place can be measured by the level of fluorescence of the dye, which is examined with a scanner. This scanner then outputs a text file for each array, which contains the relevant data pertaining to that array, such as the level of fluorescence of each spot and the level of background noise. It is these text files which are subsequently computationally analysed. In theory, a spot with brighter fluorescence means that more mRNA has hybridised, which in turn infers that more mRNA was present in sample extracted from the original cell and that the gene represented by this spot is experiencing a higher level of expression.

The types of DNA microarrays most widely used today can be broadly divided into two categories, cDNA arrays and oligonucleotide arrays.

1.2.1 The Affymetrix GeneChip

The GeneChip, which is manufactured by Affymetrix, is an oligonucleotide array and is the most commonly used type of DNA microarray [25]. They differ slightly in operation from other kinds of arrays. Each array will contain hundreds of thousands of probe spots and each of these spots will in turn contain millions of copies of an individual 25 base long DNA oligonucleotide [13].

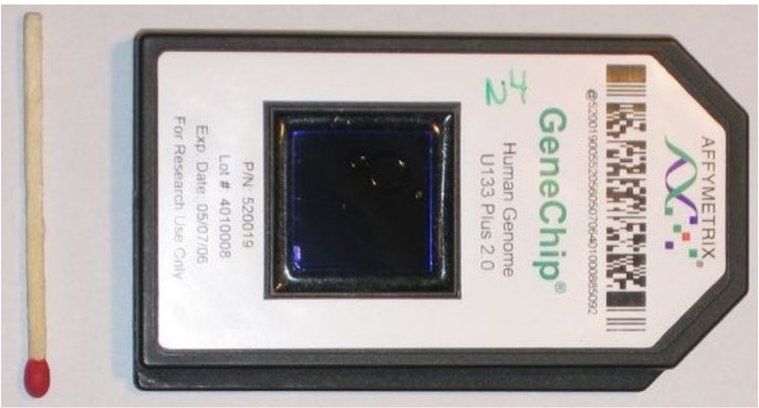


Figure 1.1: Affymetrix Microarray

Each gene that is being targeted is represented by typically (but not necessarily always) by 11 pairs of these probes. This set of probes contains 11 perfect match (PM) probes, which are exactly complementary to the DNA sequence of a subset of 25 bases of the target gene. Each PM probe has a corresponding mismatch probe (MM), which contains the same 25 base long sequence as the PM probe, except for the fact that the middle base, or the 13th base in the chain, is substituted for the complement of the 13th base of its corresponding PM probe; so for example, a G in the 13th base of a PM probe will be replaced with a C in the MM probe. This is meant to give an estimate of non-specific binding, which occurs when mRNA that is not targeted binds to a PM probe. More on how these PM and MM values are analysed to form an expression level for a gene is discussed in the next chapter.

1.2.2 cDNA Arrays

cDNA microarrays differ from Affymetrix arrays in that each spot corresponds entirely to a specific gene. Sometimes duplicate spots will target the same gene, but these spots are exact copies of each other. The probes are of varying length but are generally hundreds of bases long. Instead of mRNA levels being directly measured, these arrays measure complementary DNA (cDNA), because this is more stable molecule than mRNA at these large sizes. mRNA from the original sample is reverse transcribed in a laboratory to create an equivalent number of the more stable cDNA molecules which are then hybridised to the microarray. These cDNA molecules are usually more than 500 bases long. Each of the probes contained on the spots on the microarary will be complementary to a cDNA molecule that represents a given gene. Thus, the measure of how much cDNA binds to its corresponding spot gives an accurate measure of the expression level of the gene in question, assuming that nothing has gone wrong.

Also instead of expression levels of an individual sample being measured directly, two separate samples are hybridised to the same array at the one time. One of these samples is generally a control sample, while the other one is a sample of interest such as tumour tissue. Each of these samples is labeled with a particular dye; either a red-fluorescent dye, Cyanine 5 or Cy5, or a green-fluorescent dye, Cyanine 3 or Cy3. When the array is read by the scanner, the differential expression level of a given gene is measured by the difference in intensity level between the red and green channel, at the spot that corresponds to the gene in question.

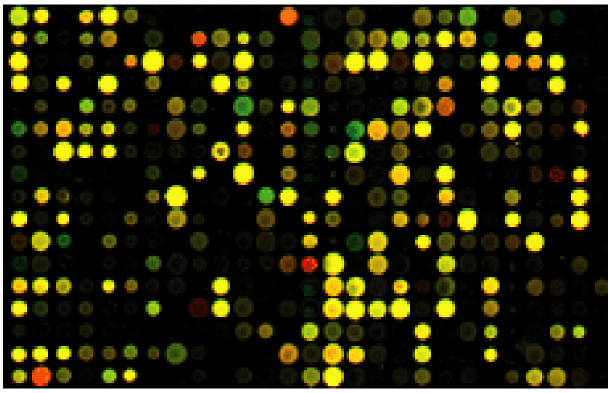


Figure 1.2: An image of a cDNA array having been read with a scanner, note the separate red and green colour channels

cDNA microarrays are initially read by a scanner, which produces a TIFF image of the array. These images are then interpreted by one of a number of image analysis software packages, all of which output data in slightly different formats. This system supports analysis of data from the major platforms, including Spotfire, GenePix, BlueFuse and Agilent.

1.2.3 miRNA Arrays

Microarrays can also be used for detection of miRNA expression levels. miRNA are short RNA molecules, generally about 22 nucleotides in length. They are encoded in genes but are not translated into proteins; instead, these molecules down regulate the expression of certain genes. They achieve this by being complementary to specific mRNA molecules created in a cell. The miRNA molecules bind to the complementary sections of these mRNAs and stop them from being converted into proteins [22].

Exiqon manafacture microarrays for detection of miRNA expression. The spots on these microarrays consist of Locked Nucleic Acid (LNA) probes. LNA is a modified RNA nucleotide that, because of the short length of miRNA molecules, forms a more stable bond with miRNA than standard DNA probes meaning that accuracy of measurements is increased [31]. The miRNA molecules that are being targeted will bind to its complementary LNA probe.

Other than this the processes of labelling the sample with a fluorescent dye, hybridising the sample to the array and reading the hybridisation levels with a scanner are similar to those of other arrays.

1.3 Thesis Structure

The abstract and this chapter have served to introduce the basic biological concepts underlying the working of microarrays as well as outlining the reason for this project and its aims.

- Chapter 2 aims to outline some of the most common techniques used in analysing microarray data and gives an outline of several possible analysis pipelines.
- Chapter 3 provides a detailed description of the technologies used in the project as well as justification as to why certain tools were chosen.
- Chapter 4 provides a detailed manual on how the system is used to analyse actual microarray data.
- Chapter 5 details information describing how the various components of the system fit together and how various parts of the application are constructed, communicate and operate.
- Chapter 6 focuses on the limitations of a systems like this, what recommendations would be made for future work this area and where the potential is for future expansion of this project.

Chapter 2

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.

2.1 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.

2.1.1 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 [20]. 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 [7]. 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 [10].

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 [5].

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 [25]. 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.

2.1.2 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) [2]. 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. Remeber than approximently 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 [1]. 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 [11].

The methods works 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) [5]. Robust multi-array analysis is then carried out on the quantiles [11].

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 [30].

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

  - Contrasts
  - Invariant Set
  - Loess
  - Qspline
  - Quantiles
  - Robust Quantiles

- PM Correction:
  - PM Only
  - Subtract MM
  - MAS5

- Summarisation:
  - Average Difference
  - LiWong
  - MAS5
  - Median Polish
  - Playerout

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

2.1.3 Preprocessing of cDNA Data

The general steps followed when preprocessing cDNA data are quite similar to the above. The main differences are that their is no need for PM correction, as there are no MM probes on cDNA arrays and that their 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 [20]. 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. [27]

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.

2.1.4 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 [20]:
    - Subtract
    - Half
    - Minimum
    - MovingMin
    - Edwards
    - NormExp
    - RMA
- Normalize Between Arrays [27]:
    - Aquantile

  - Scale
  - Quantile
  - Gquantile
  - Rquantile
  - Tquantile

- Normalize Within Arrays[27]:
  - Print Tip Loess
  - Median
  - Loess
  - Composite
  - Control
  - Robust Spline

2.1.5 Preprocessing of Single Dye Arrays

The VSN method [9] 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

$$x_{ki} \mapsto h_i(x_{ki}) = \text{glog}\left(\frac{x_{ki} - a_i}{b_i}\right) \tag{2.1}$$

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) [21].

2.2 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 aides 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.

2.2.1 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.

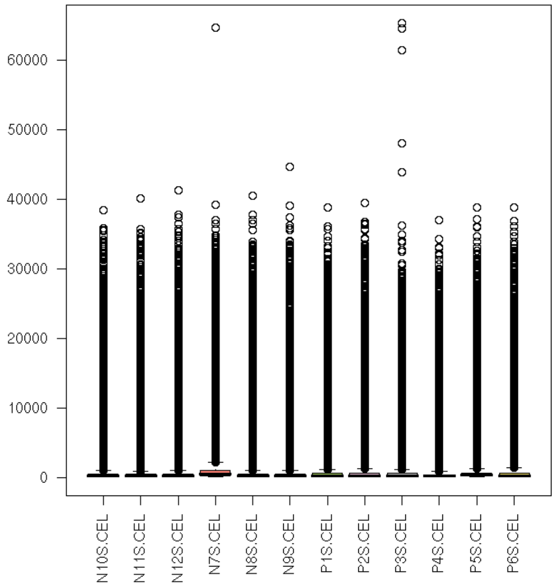


Figure 2.1: Boxplot of raw probe intensity values

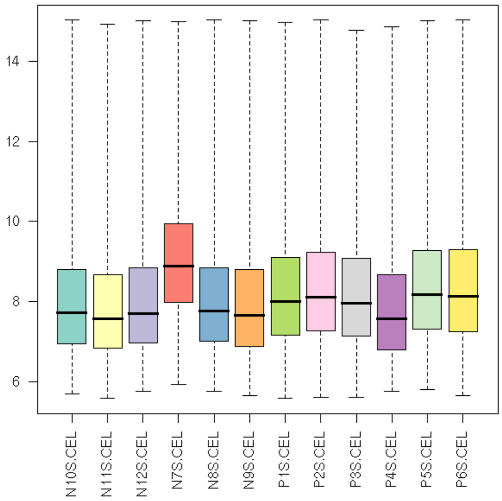


Figure 2.2: Boxplot of log_2 transformed probe intensity values

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 preform 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.

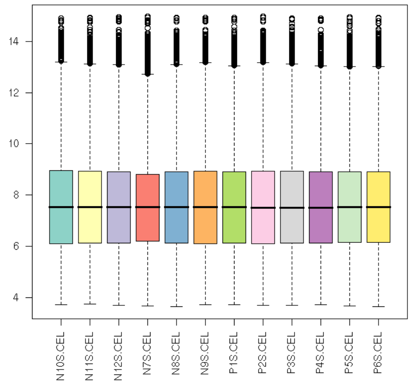


Figure 2.3: RMA Preprocessed Intensities

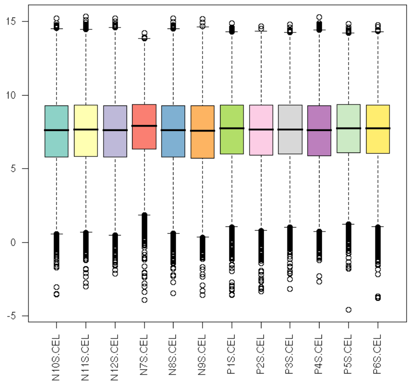


Figure 2.4: MAS5 Preprocessed Intensities

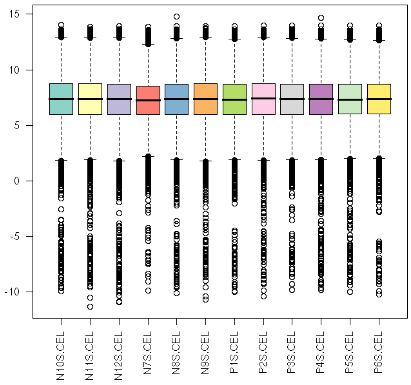


Figure 2.5: PLIER Preprocessed Intensities With Quantile Normalization

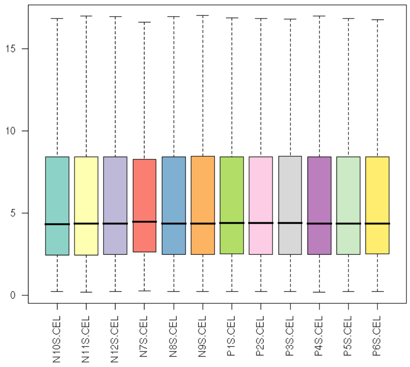


Figure 2.6: GCRMA Preprocessed Intensities

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.

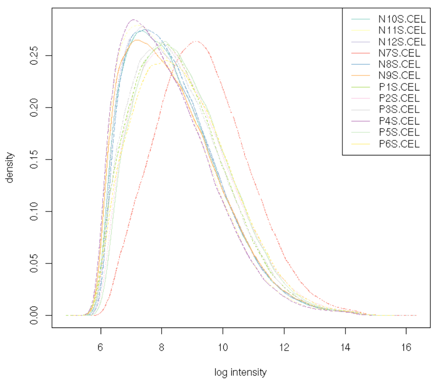


Figure 2.7: Raw probe intensity values plotted on a histogram

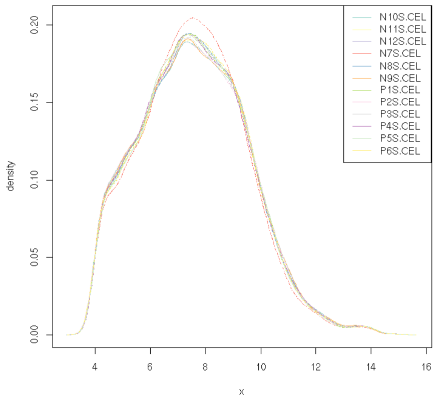


Figure 2.8: RMA Preprocessed Histogram

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.

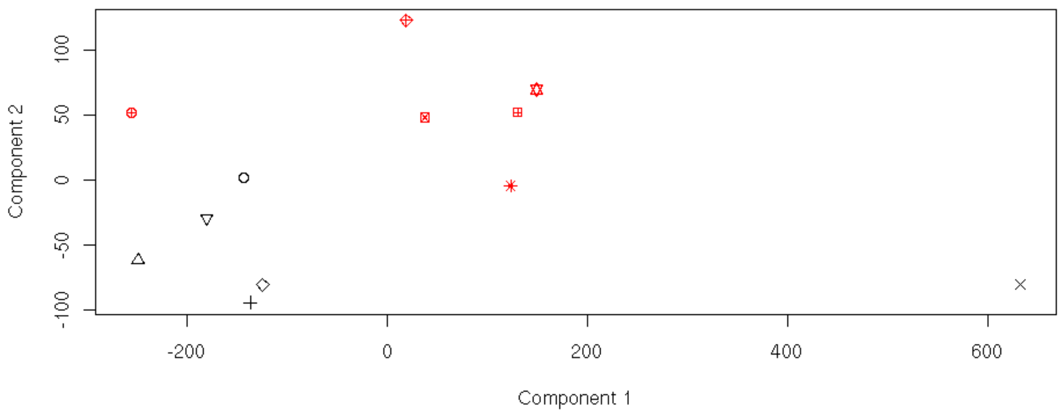


Figure 2.9: PCA plot of raw data. Note non-grouping of NS7.CEL.

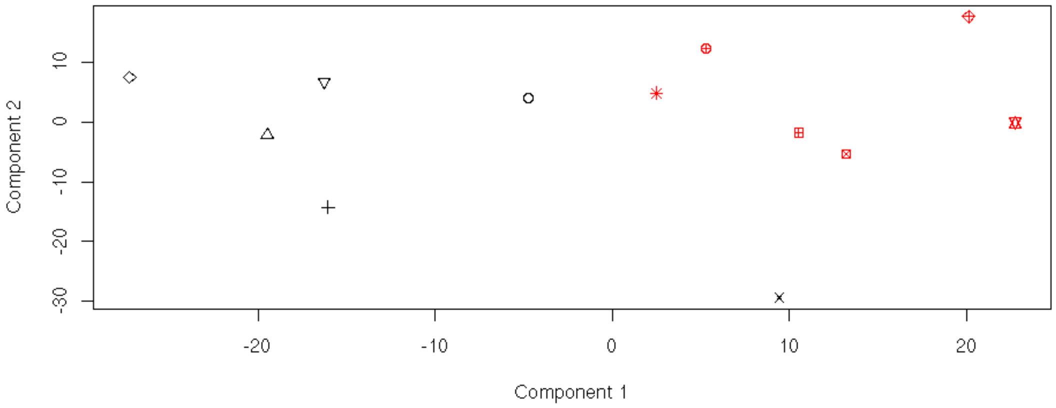


Figure 2.10: PCA plot of RMA preprocessed data. Note improved overall clustering but continued non-grouping of NS7.CEL.

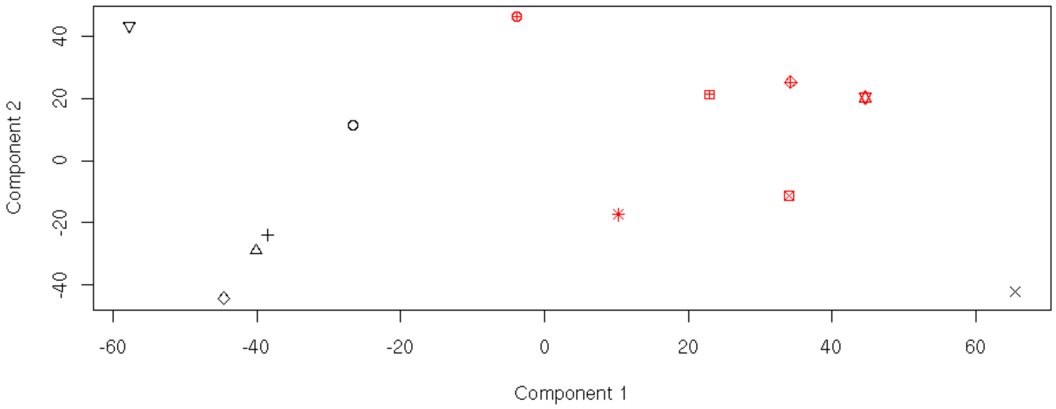
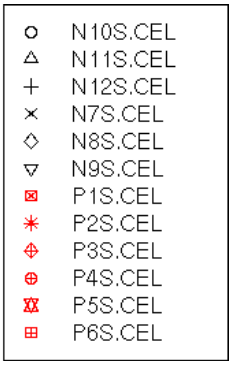


Figure 2.11: PCA plot of MAS5 preprocessed data. Once again the situation is not ameliorated for NS7.CEL



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 [4].

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.

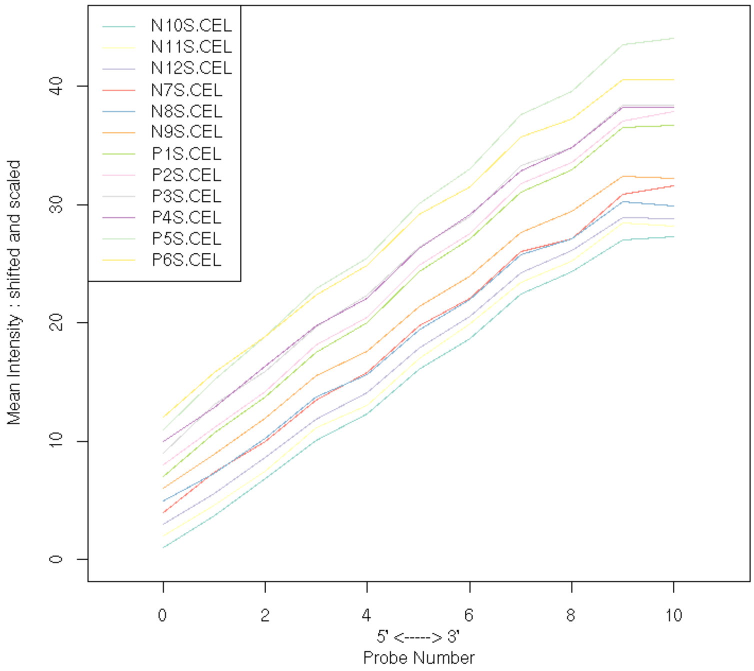


Figure 2.12: RNA Degradation Plot

Simple Affy Plot and Affymetrix Recommended Metrics

Affymetrix reccomends 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 [29].

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 chips overall values are scaled because of this. Affymetrix recommends that scale factors be within 3-fold of each other [29].

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 [29]. Again it is recommended by Affymetrix that these values be similar.

Below are tables of these values for our example dataset.

Table 2.1: Background Levels

N12S.CEL	N11S.CEL	N8S.CEL	P1S.CEL	P4S.CEL	P5S.CEL
67.49	62.04	68.94	62.50	60.89	73.18
N10S.CEL	P3S.CEL	P6S.CEL	P2S.CEL	N7S.CEL	N9S.CEL
65.61	64.53	64.25	70.25	84.85	62.75

Table 2.2: Scale Factors

N12S.CEL	N11S.CEL	N8S.CEL	P1S.CEL	P4S.CEL	P5S.CEL
0.44	0.50	0.45	0.26	0.44	0.43
N10S.CEL	P3S.CEL	P6S.CEL	P2S.CEL	N7S.CEL	N9S.CEL
0.40	0.33	0.38	0.51	0.36	0.34

Table 2.3: Percent Present Calls

N12S.CEL 57.73%	N11S.CEL 58.61%	N8S.CEL 57.69%	P1S.CEL 57.89%	P4S.CEL 57.75%	P5S.CEL 58.16%
N10S.CEL 58.97%	P3S.CEL 59.57%	P6S.CEL 58.17%	P2S.CEL 59.46%	N7S.CEL 54.78%	N9S.CEL 58.90%

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.

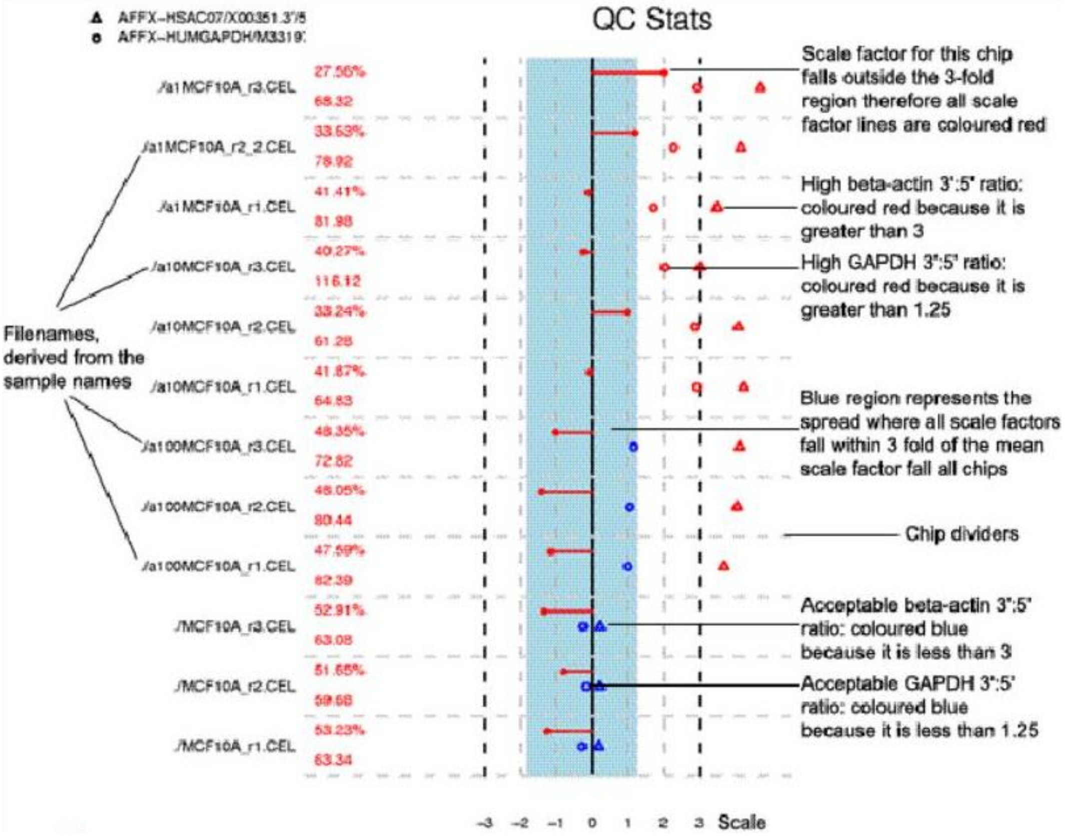


Figure 2.13: SimpleAffy Plot

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) [8] and implemented in R by the rlm() function from the package MASS by Venables and Ripley (2002) [28]. This will be further discussed in chapter 3.

$$log(Y_{gij}) = \theta_{gi} + \phi_{gj} + \varepsilon_{gij} \tag{2.2}$$

The above equation is referred to as a probe level model; θ_{gi} represents the log transformed expression level for gene g on array i, ϕ_{gj} is the effect of the j-th probe representing gene i, and ε_{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 [4].

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 articfacts 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 ϕ, is not obvious in the image of probe intensities.

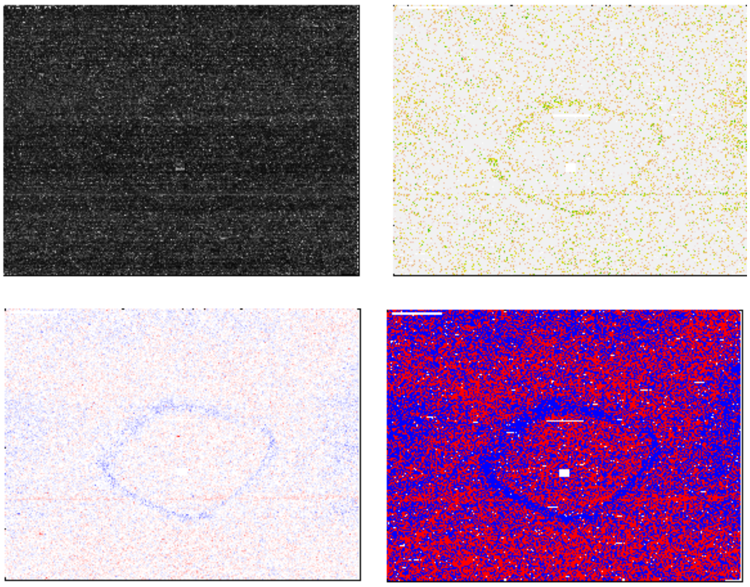


Figure 2.14: Array Images

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 θ_{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 $M = 0$ [4].

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 [4].

$$\text{NUSE}(\theta_{gi}) = \frac{\text{SE}(\theta_{gi})}{\text{med}_i(\text{SE}(\theta_{gi}))} \tag{2.3}$$

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.

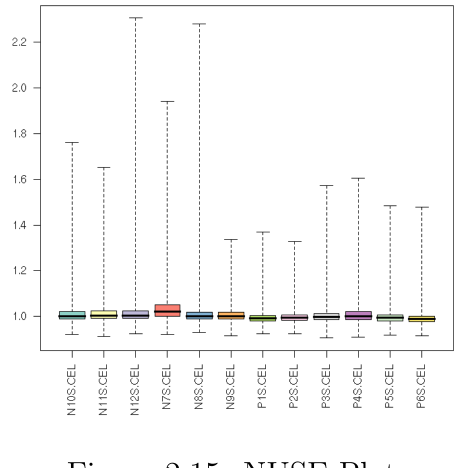


Figure 2.15: NUSE Plot

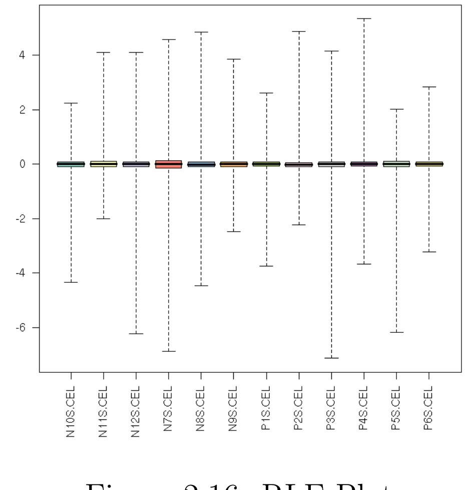


Figure 2.16: RLE Plot

2.2.2 Quality Assessment of cDNA Data

There are also number of useful tools implemented to assess the quality of cDNA data. This subsection aims to 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

$$A = \log_2 \sqrt{Cy5 \cdot Cy3} = \frac{1}{2}[\log_2(Cy5) + \log_2(Cy3)] \tag{2.4}$$

Cy5 and Cy3 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

$$M = \log_2 \frac{Cy5}{Cy3} = \log_2 \frac{Cy5}{Cy3} = \log_2(Cy5) - \log_2(Cy3) \tag{2.5}$$

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.

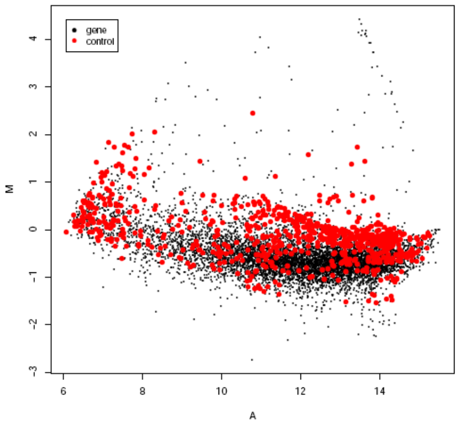


Figure 2.17: MA-plot of Raw Swirl.1.spot

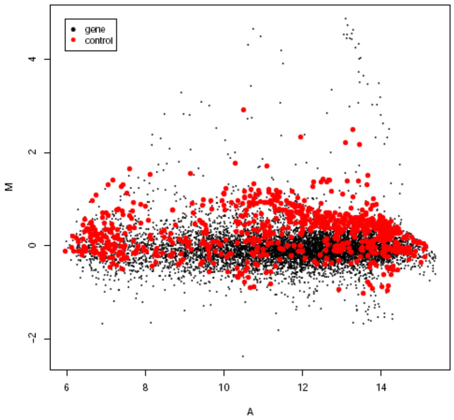


Figure 2.18: MA-plot of print-tip-group loess normalized Swirl.1.spot

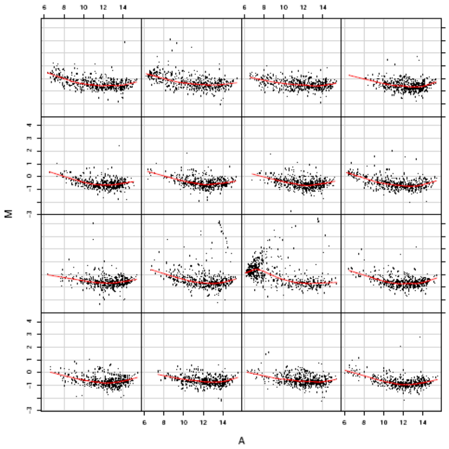


Figure 2.19: MA plots for raw print tip groups of Swirl.1.spot

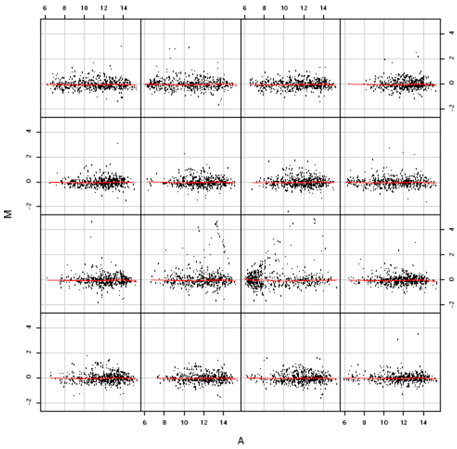


Figure 2.20: MA plots of normalized print tip groups of raw Swirl.1.spot

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.

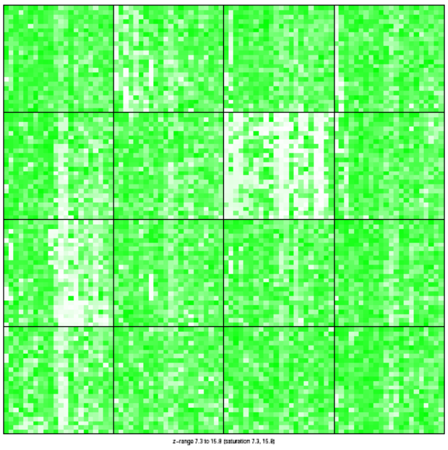


Figure 2.21: Array image of green foreground intensities

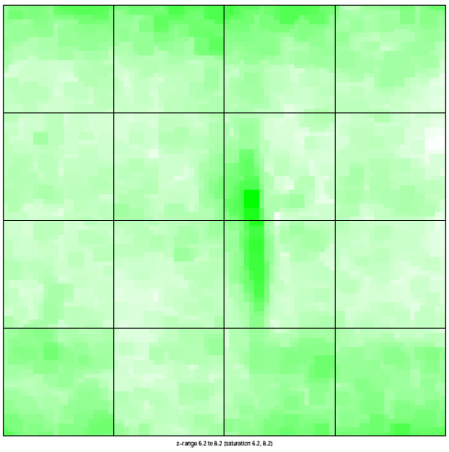


Figure 2.22: Array image of green background intensities

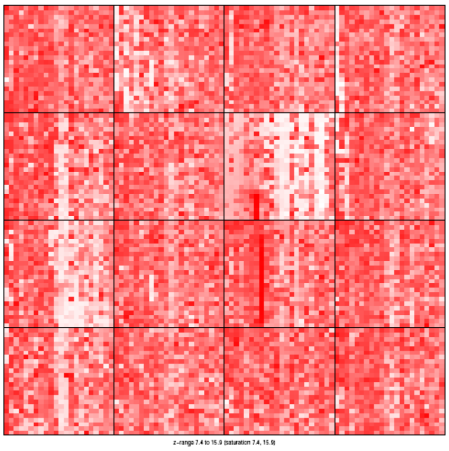


Figure 2.23: Array image of red foreground intensities

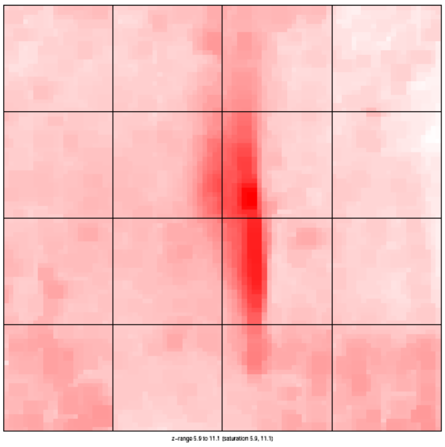


Figure 2.24: Array image of red background intensities

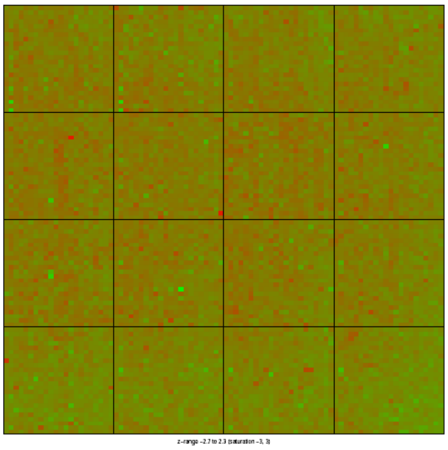


Figure 2.25: Array images of M (log-ratios) of raw data

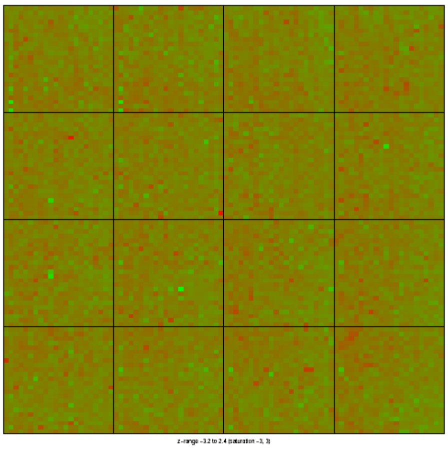


Figure 2.26: Array images of M (log-ratios) of normalized data

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. 2.27), 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.

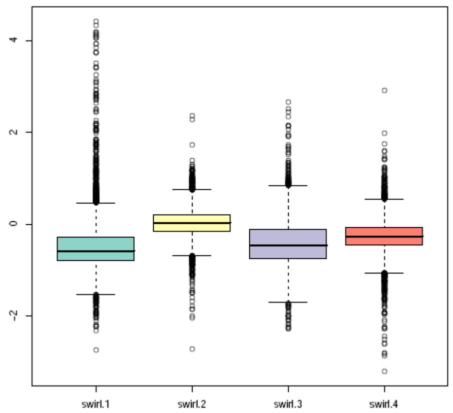


Figure 2.27: Boxplot of M for raw data

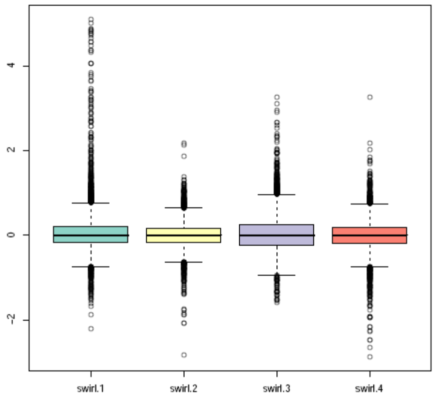


Figure 2.28: Boxplot of M for quantile normalized data

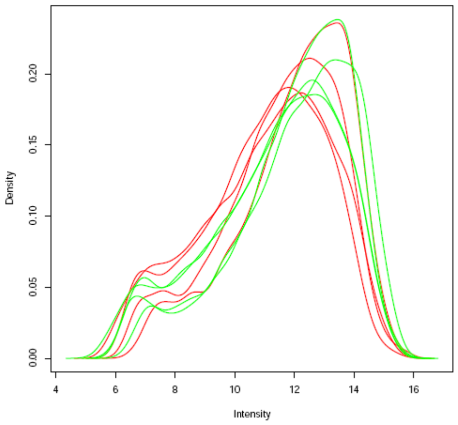


Figure 2.29: Raw Intensities Plot

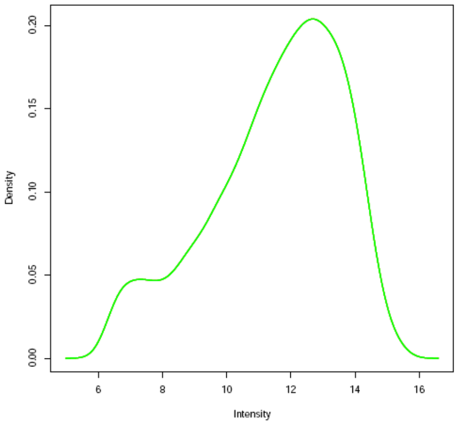


Figure 2.30: Quantile Normalized Intensities Plot

2.2.3 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.

2.3 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 [24].

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 ∼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) [3], 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.

2.3.1 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.

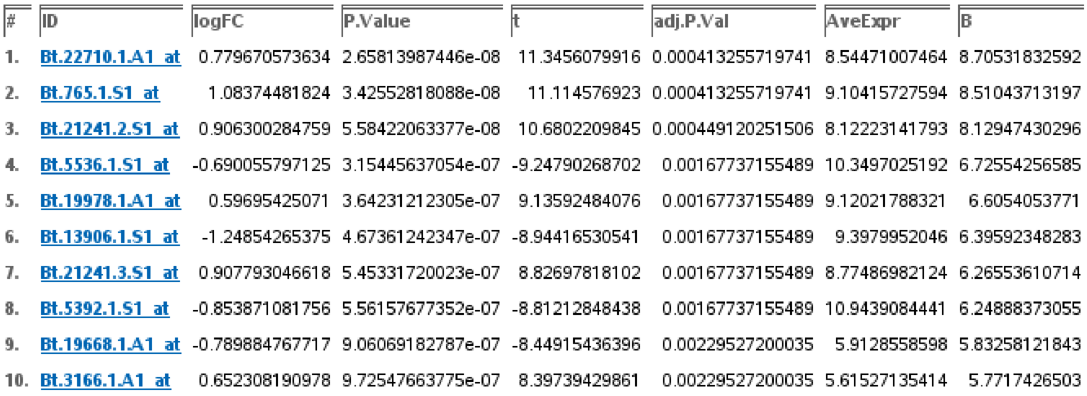
This procedure first fits a linear model to the expression data for each probe. The expression data should be log-ratios M for two-colour array platforms or log-expression values for one-channel platforms. The coefficients of the fitted models describe the differences between the RNA sources hybridised 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 [26].

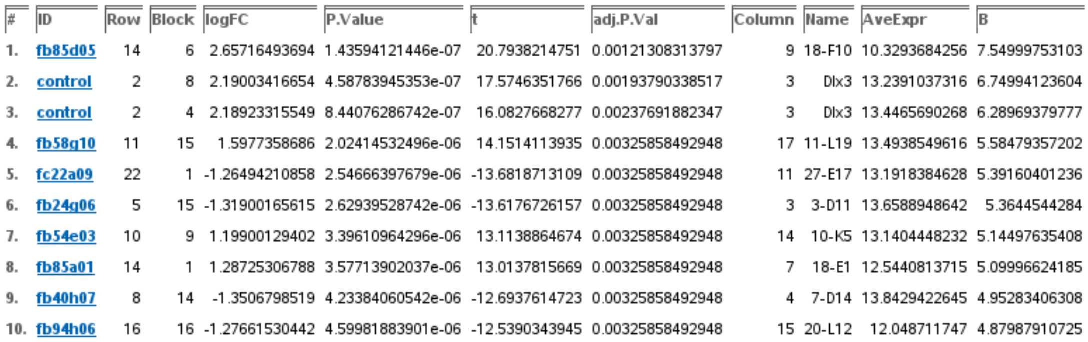
Below are screen shots of the top ranked differentially expressed genes from the two datasets we introduced earlier. The GeneChip data (bovine dataset) was preprocessed using RMA; while the Swirl dataset, which is a dual dye experiment where array image analysis was performed using Spotfire Software, 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.



#	ID	logFC	P.Value	t	adj.P.Val	AveExpr	B
1.	Bt.22710.1.A1_at	0.779670573634	2.65813987446e-08	11.3456079916	0.000413255719741	8.54471007464	8.70531832592
2.	Bt.765.1.S1_at	1.08374481824	3.42552818088e-08	11.114576923	0.000413255719741	9.10415727594	8.51043713197
3.	Bt.21241.2.S1_at	0.906300284759	5.58422063377e-08	10.6802209845	0.000449120251506	8.12223141793	8.12947430296
4.	Bt.5536.1.S1_at	-0.690055797125	3.15445637054e-07	-9.24790268702	0.00167737155489	10.3497025192	6.72554256585
5.	Bt.19978.1.A1_at	0.59695425071	3.64231212305e-07	9.13592484076	0.00167737155489	9.12021788321	6.6054053771
6.	Bt.13906.1.S1_at	-1.24854265375	4.67361242347e-07	-8.94416530541	0.00167737155489	9.3979952046	6.39592348283
7.	Bt.21241.3.S1_at	0.907793046618	5.45331720023e-07	8.82697818102	0.00167737155489	8.77486982124	6.26553610714
8.	Bt.5392.1.S1_at	-0.853871081756	5.56157677352e-07	-8.81212848438	0.00167737155489	10.9439084441	6.24888373055
9.	Bt.19668.1.A1_at	-0.789884767717	9.06069182787e-07	-8.44915436396	0.00229527200035	5.9128558598	5.83258121843
10.	Bt.3166.1.A1_at	0.652308190978	9.72547663775e-07	8.39739429861	0.00229527200035	5.61527135414	5.7717426503

Figure 2.31: Top 10 genes for Bovine dataset



#	ID	Row	Block	logFC	P.Value	t	adj.P.Val	Column	Name	AveExpr	B
1.	fb85d05	14	6	2.65716493694	1.43594121446e-07	20.7938214751	0.00121308313797	9	18-F10	10.3293684256	7.54999753103
2.	control	2	8	2.19003416654	4.58783945353e-07	17.5746351766	0.00193790338517	3	Dlx3	13.2391037316	6.74994123604
3.	control	2	4	2.18923315549	8.44076286742e-07	16.0827668277	0.00237691882347	3	Dlx3	13.4465690268	6.28969379777
4.	fb58g10	11	15	1.5977358686	2.02414532496e-06	14.1514113935	0.00325858492948	17	11-L19	13.4938549616	5.58479357202
5.	fc22a09	22	1	-1.26494210858	2.54666397679e-06	-13.6818713109	0.00325858492948	11	27-E17	13.1918384628	5.39160401236
6.	fb24g06	5	15	-1.31900165615	2.62939528742e-06	-13.6176726157	0.00325858492948	3	3-D11	13.6588948642	5.3644544284
7.	fb54e03	10	9	1.19900129402	3.39610964296e-06	13.1138864674	0.00325858492948	14	10-K5	13.1404448232	5.14497635408
8.	fb85a01	14	1	1.28725306788	3.57713902037e-06	13.0137815669	0.00325858492948	7	18-E1	12.5440813715	5.09996624185
9.	fb40h07	8	14	-1.3506798519	4.23384060542e-06	-12.6937614723	0.00325858492948	4	7-D14	13.8429422645	4.95283406308
10.	fb94h06	16	16	-1.27661530442	4.59981883901e-06	-12.5390343945	0.00325858492948	15	20-L12	12.048711747	4.87987910725

Figure 2.32: Top 10 genes for Swirl dataset

2.3.2 The PUMA Package

PUMA is an acronym for Propagating Uncertainty in Microarray Analysis [19]. 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 ∼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[15], which uses Bayesian methods to associate credibility intervals with expression levels.

For PCA, a noise-propagation in principal components analysis method[23] 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[16] 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 [19].

2.4 Use of Remapped Probe Sets For GeneChip Arrays

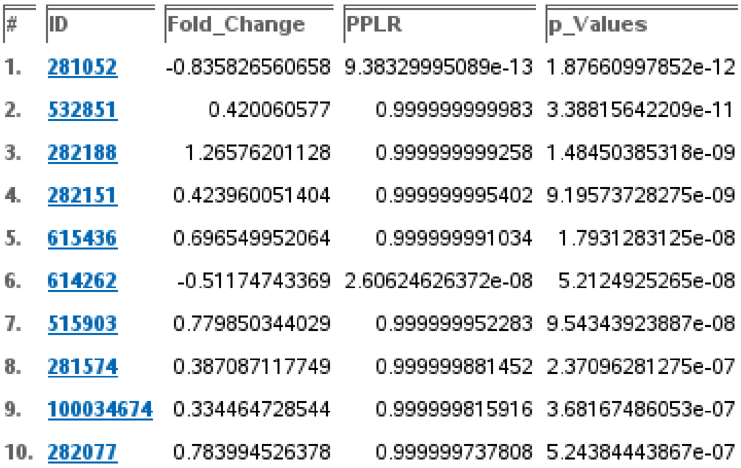
As already outlined, most GeneChip arrays use 11 different 25-base long probes to target specific genes.

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 [14], 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 therefor provide a more reliable measure of a genes true expression level[6].

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.



#	ID	Fold_Change	PPLR	p_Values
1.	281052	-0.835826560658	9.38329995089e-13	1.87660997852e-12
2.	532851	0.420060577	0.999999999983	3.38815642209e-11
3.	282188	1.26576201128	0.999999999258	1.48450385318e-09
4.	282151	0.423960051404	0.999999995402	9.19573728275e-09
5.	615436	0.696549952064	0.999999991034	1.7931283125e-08
6.	614262	-0.51174743369	2.60624626372e-08	5.2124925265e-08
7.	515903	0.779850344029	0.999999952283	9.54343923887e-08
8.	281574	0.387087117749	0.999999881452	2.37096281275e-07
9.	100034674	0.334464728544	0.999999815916	3.68167486053e-07
10.	282077	0.783994526378	0.999999737808	5.24384443867e-07

Figure 2.33: Top 10 genes for Bovine dataset (PUMA)

Chapter 3

Using the System

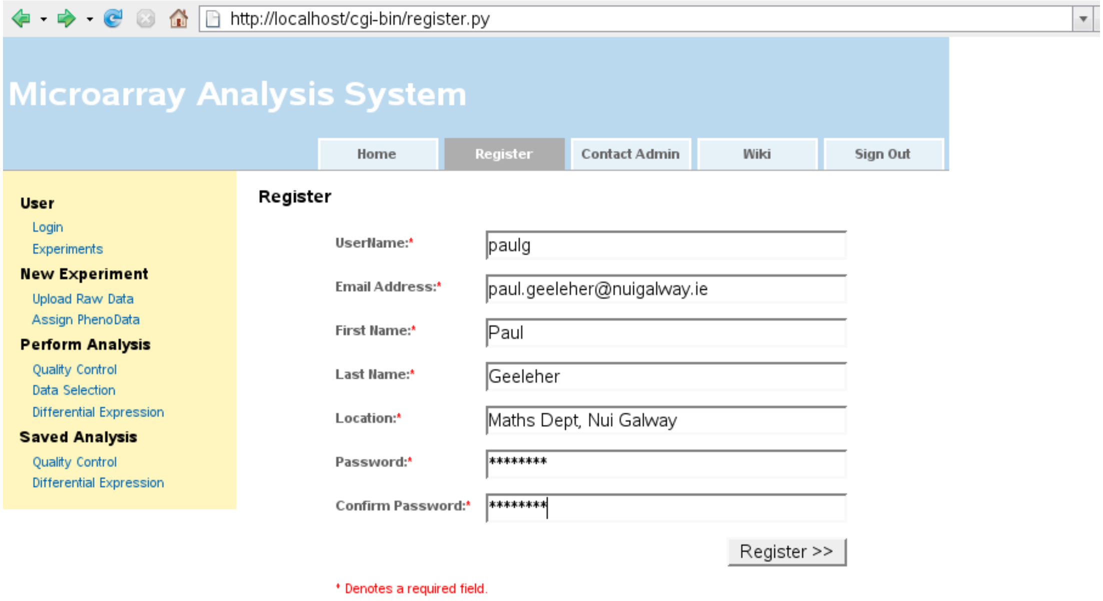
3.1 Overview

In this chapter I will outline some of the analysis pipelines that can be followed to go from several different types of raw data to annotated lists of differentially expressed genes. The example datasets I will use are the Bovine and Swirl datasets that I introduced earlier, as well as the Estrogen dataset, which is packaged with Bioconductor

3.2 Registering and Uploading Your First Experiment

The first thing a user must do is register with the system. This is achieved by clicking the “Register” tab on the top of the page which invokes the “register.py” script and handles this process. This page contains several JavaScript functions to authenticate that the form has been filled out correctly, verifying for example that the email address supplied is valid and all required fields have been completed. If the forum is not properly filled out, an error dialogue box will inform the user and prevent submission of the page.

Once registration is complete the user may now proceed to login for the first time. See the figures on the next page for details.



Register

UserName:* paulg

Email Address:* paul.geeleher@nuigalway.ie

First Name:* Paul

Last Name:* Geeleher

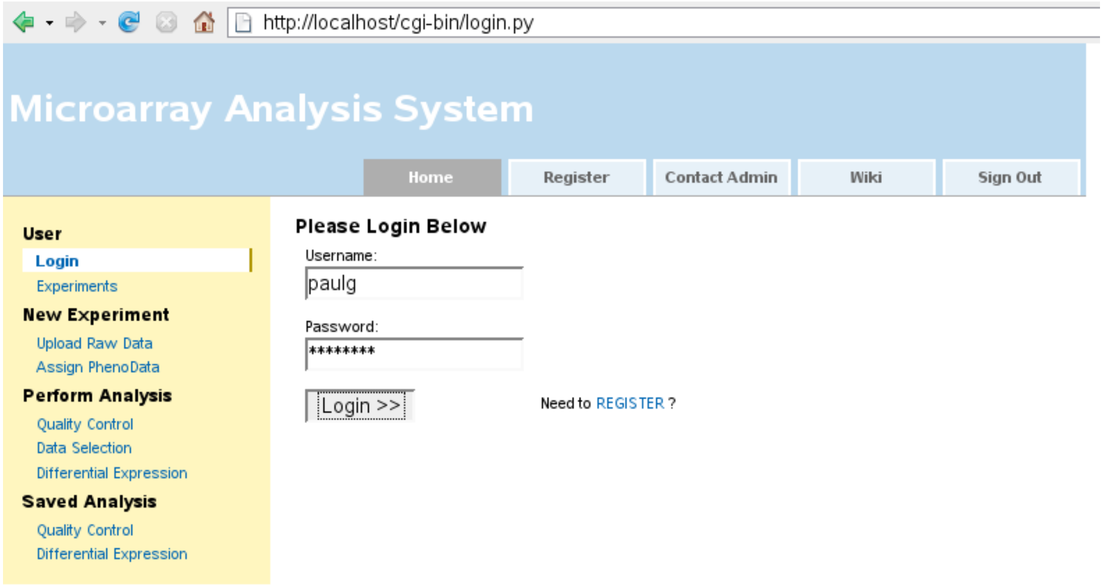
Location:* Maths Dept, Nui Galway

Password:* ********

Confirm Password:* ********

* Denotes a required field.

Figure 3.1: Registration Screen



Please Login Below

Username: paulg

Password: ********

Need to REGISTER ?

Figure 3.2: Login Screen

3.3 Uploading Data

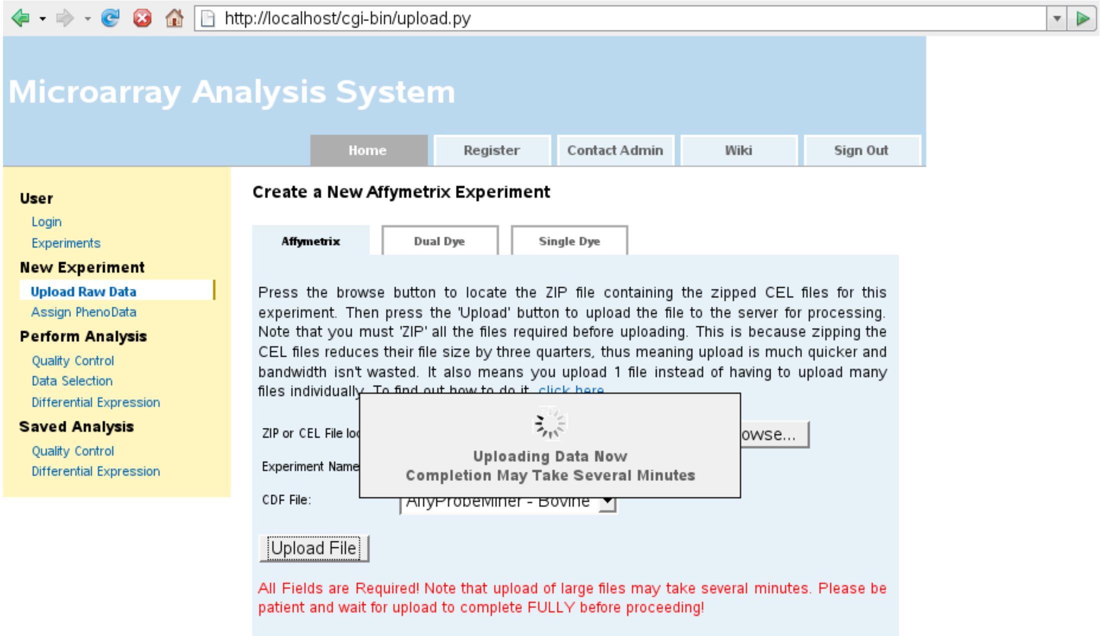
The first stage in creating an experiment is to upload the raw data. At this stage the user is presented with a page that displays options to upload 3 different types of data, Affymetrix, dual dye or single dye.

For all data types the user must package their raw data files in a zip archive. This can be accomplished with any number of freely available compression tools like 7-Zip or WinZip. Zipping the data has the double advantage of significantly compressing the data for faster upload and also means that only one file needs to be uploaded. The user must also specify an experiment name every time they upload a new dataset. There are some differences in the rest of the information supplied when uploading different types of experiments. For Affymetrix data, the user has the choice of using the default Affymetrix supplied CDF file or using the remapped CDF file from AffyProbeMiner. This was discussed in Chapter 2. For Dual dye data, the user may wish to specify a “Spot Types” file. This file contains information which describes the function of certain spots on the arrays and can be used to identify, for example, control signals. Depending on the type of data being uploaded, a GenePix Array List (GAL) file may also need to be specified. This file contains information pertaining to the physical layout of the arrays used in the experiment and is needed to generate array images, as well as for certain kinds of normalisation. The information contained in a GAL file can usually be read from the actual raw data files themselves, so this field is optional. The next required field designates the image analysis program that was used to produce the files being uploaded. The options available are “Spot”, “GenePix”, “BlueFuse” and “Agilent”. Single dye experiments have all of the same upload options as dual dye (including GAL and Spot Types files), except for the fact that the names of the columns containing foreground and background intensity levels in the raw data files can also be specified.

Given the large amount of data that can be involved in microarray analysis and the often lengthly delays that are experience in processing data, one of the key usability issues of this system is that the user is never presented with a blank or static screen for any amount of time and always remains informed on what is happening on the systems back end [17]. This is particularly important where a web based system is concerned, as it is all too tempting for a user to click a browser’s stop or back button if they get even the slightest feeling something has gone wrong, which will of course halt execution of whatever is happening. This issue has been addressed in every aspect of this system where delay may be involved.

Our bovine dataset, which includes 12 arrays, weighs in at a hefty 41 megabytes, even when compressed. This upload will take several minutes, even on a quick connection.

The first screenshot on the next page shows the screens directly after the upload button has been clicked. This is what is seen while the actual data transfer is in progress. Note how the user is informed to be patient and told that the upload will take several minutes, while the spinning animation gives the all important impression that something is happening. The next screen shows what has happened after data is uploaded. For this dataset this screen takes about 30 seconds to work to completion. Crucially, progress is printed as each step is completed, so the user is not presented with a static screen until the page is fully loaded.



Create a New Affymetrix Experiment

Press the browse button to locate the ZIP file containing the zipped CEL files for this experiment. Then press the 'Upload' button to upload the file to the server for processing. Note that you must 'ZIP' all the files required before uploading. This is because zipping the CEL files reduces their file size by three quarters, thus meaning upload is much quicker and bandwidth isn't wasted. It also means you upload 1 file instead of having to upload many files individually. To find out how to do it, click here

ZIP or CEL File loc

owse...

Experiment Name

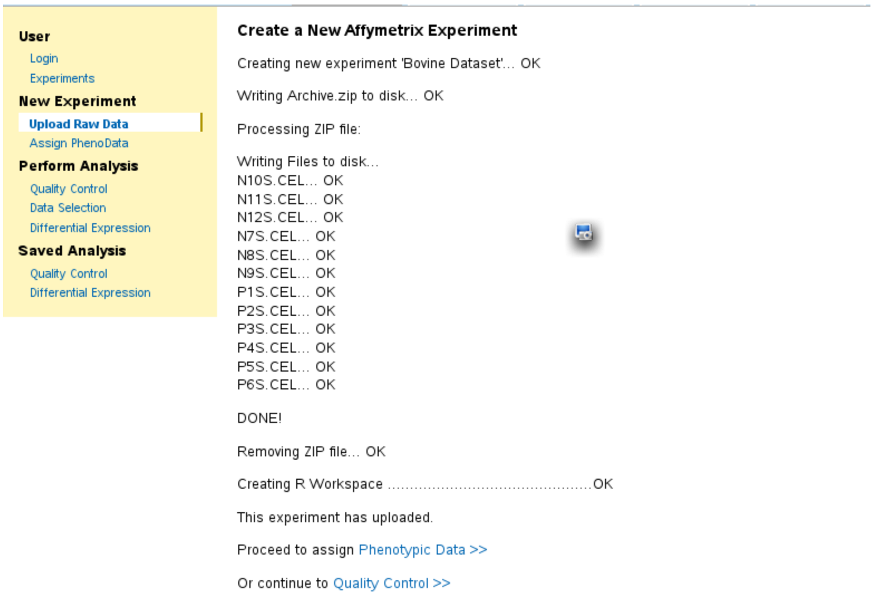
CDF File: AffyProbeMiner - Bovine

Uploading Data Now
Completion May Take Several Minutes

Upload File

All Fields are Required! Note that upload of large files may take several minutes. Please be patient and wait for upload to complete FULLY before proceeding!

Figure 3.3: The upload screen after the upload button is clicked.



Create a New Affymetrix Experiment

Creating new experiment 'Bovine Dataset'... OK

Writing Archive.zip to disk... OK

Processing ZIP file:

```
Writing Files to disk...
N10S.CEL... OK
N11S.CEL... OK
N12S.CEL... OK
N7S.CEL... OK
N8S.CEL... OK
N9S.CEL... OK
P1S.CEL... OK
P2S.CEL... OK
P3S.CEL... OK
P4S.CEL... OK
P5S.CEL... OK
P6S.CEL... OK
```

DONE!

Removing ZIP file... OK

Creating R Workspace ....OK

This experiment has uploaded.

Proceed to assign Phenotypic Data >>

Or continue to Quality Control >>

Figure 3.4: The upload screen after data transfer (page heading removed from image).

3.4 Assigning Phenotypic Data

The next stage of almost any analysis is to assign the experiment's phenotypic data.

This stage differs significantly for Affymetrix datasets, where the user has the option to specify an experimental design of up to 10 factors, with up to 10 levels of each factor. To clarify what I mean by this, consider our bovine dataset from earlier. The simplest experiments contain only one factor, our bovine experiment for example contains one factor, which we could call "Negative Energy Balance"; this factor is then said to have two levels, one of which is negative energy balance group and the other of which is a control group. An appropriate level of each factor is then assigned to each array to describe the phenotypic state of the sample. In the above example, arrays will either be designated as being from the negative energy balance group or from the control group, based on which of either level of the single factor they are assigned.

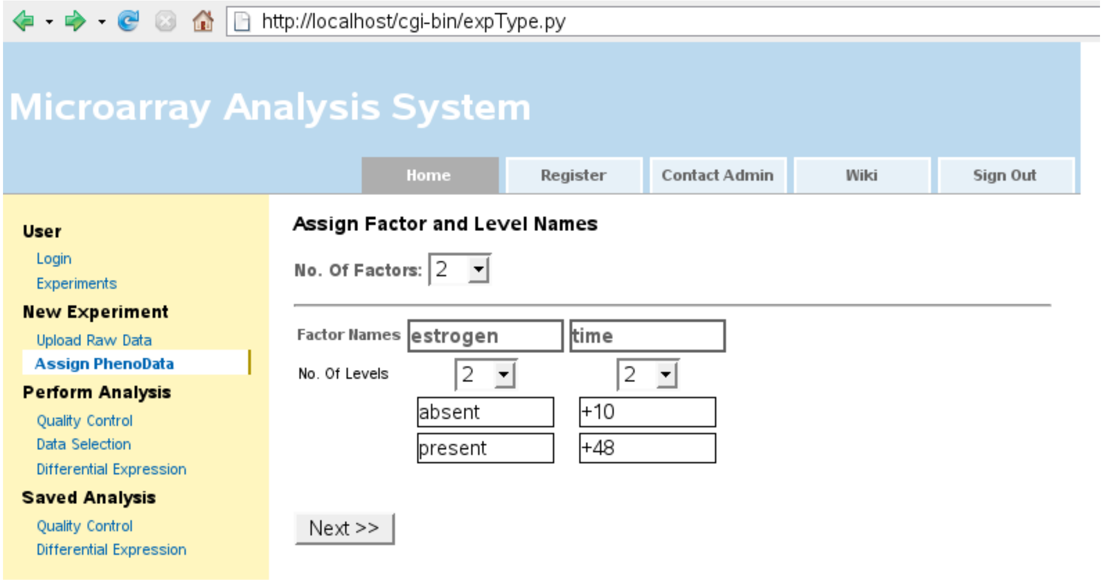
We will now consider the more complex experimental design of the Estrogen Dataset, which is one of several sample datasets that is bundled with Bioconductor. It is from an experiment on MCF7 human breast cancer cells using Affymetrix HGU95av2 arrays. The aim of the study was to identify genes which respond to estrogen and to classify these into early and late responders. This experiment is of the popular 2x2 factorial design. It contains two factors, both of which have two levels. The first factor (which we will call "Estrogen") defines two different kinds of samples, which have either estrogen absent or present. We will call the two levels of this factor "Absent" and "Present". The next factor (which we will call "time") defines the length of exposure of the samples, either 10 or 48 hours, we will call these levels "+10" and "+48".

Assignment of this phenotypic data to each of the arrays allows us to define different contrasts to assess for differential expression analysis.

Fig. 3.5 on the next page shows the first stage where the user specifies the number of factors and levels required and gives them names. When the numeric values in the drop down menus that define the amount of factors or levels is changed, more text boxes are dynamically added without reloading the page. This is achieved using JavaScript.

The second screenshot (Fig. 3.6) shows the next stop, where the various levels of each factor are assigned to the arrays. When this form is submitted, the database and the R environment are updated, which allows the system to subsequently calculate which contrasts are available in differential expression analysis.

At present, single and dual dye experiments only support experimental designs that involve two RNA samples, i.e. two levels of one factor.

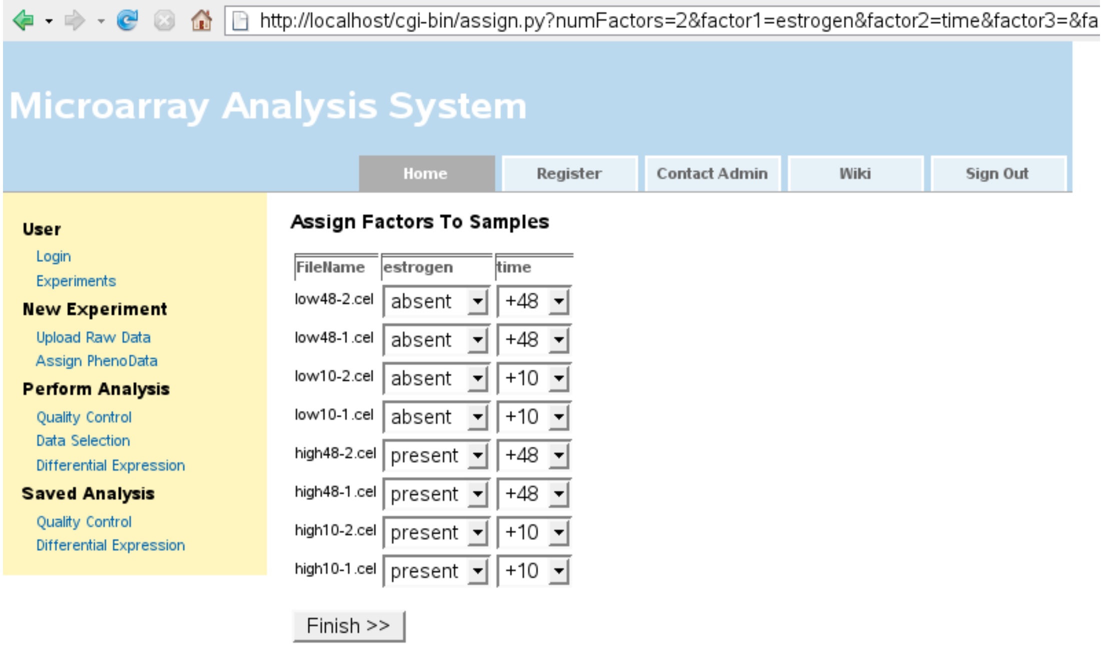


Assign Factor and Level Names

No. Of Factors: 2

Factor Names	estrogen	time
No. Of Levels	2	2
	absent	+10
	present	+48

Figure 3.5: Assigning factor and level names.



Assign Factors To Samples

FileName	estrogen	time
low48-2.cel	absent	+48
low48-1.cel	absent	+48
low10-2.cel	absent	+10
low10-1.cel	absent	+10
high48-2.cel	present	+48
high48-1.cel	present	+48
high10-2.cel	present	+10
high10-1.cel	present	+10

Figure 3.6: Assigning phenotypic data

3.5 Quality Control Analysis

The next step in any analysis is to run quality control. The actual quality control process and the plots and metrics that have been implemented were discussed in chapter 2. The quality control page allows the generation of this data.

The initial screen (Fig. 3.7) displays a list of available quality control options (boxplot, histogram etc.) that be selected using checkboxes. The options can be used to asses raw, preprocessed data, or both.

If working with dual dye or Affymetrix data, hovering the mouse pointer over the "Preprocessed Data" heading on the table (see figures on next page) allows the user to select from a number of different preprocessing methods. Changing this value will change how data is preprocessed for the appropriate plots.

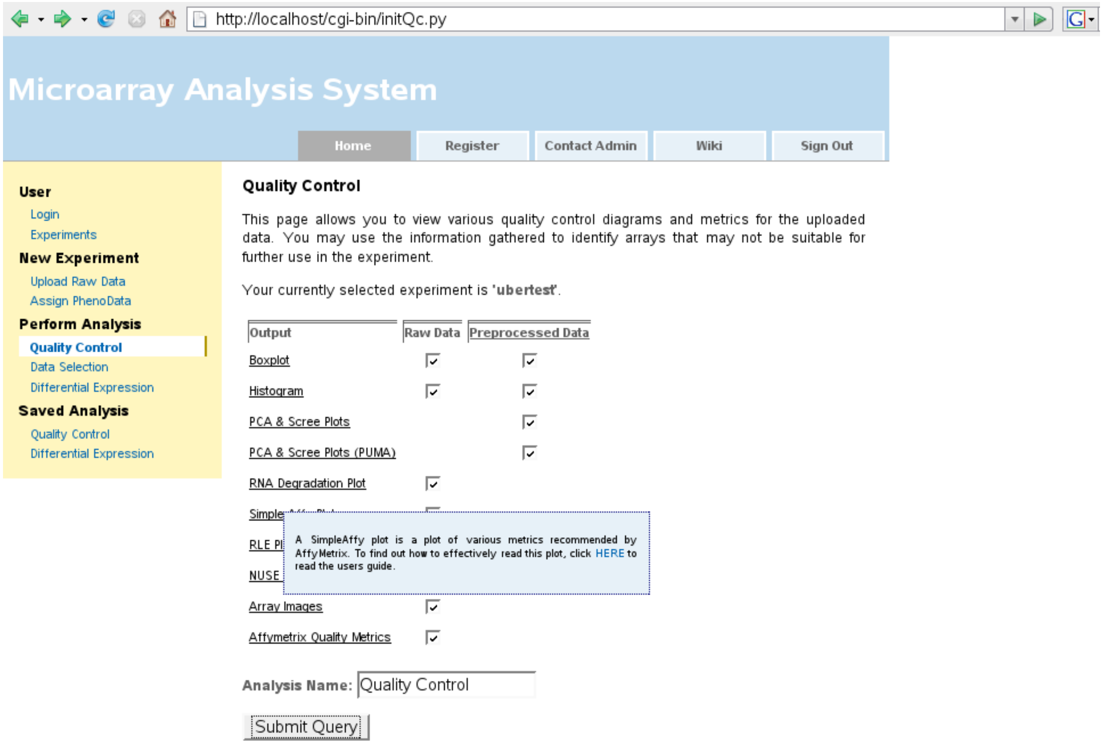
Hovering the mouse pointer over any particular quality control option will cause a tooltip to appear that gives an outline of how that particular plot or metric can be used and how it should be interpreted. This is a useful feature for novice users.

Once the required options have been checked the user will submit this form.

The next page displays everything that has been requested. Quality plots are displayed as thumbnail images, full size versions of which can be viewed by clicking the thumbnail. Any metrics such as average background or scale factors are displayed in tabular form.

It is again important to note that the output of this page is piped directly to the screen as the page is generated. If every option is checked the whole page takes about a minute to print to completion for our bovine dataset, but the user is never left staring at a static or blank screen. Even the progress of any preprocessing method that is being undertaken is piped directly from R to the users screen.

Additionally, if analysing an Affymetrix experiment and the user selects the "PUMA PCA and Scree Plot" option, instead of these plots being generated there and then, the user will instead be informed that notification of completion of the plots will be emailed to them. Due to the higher volume of data the PUMA method deals with, this plot takes a long time to generate, approximently 40 minutes for our bovine dataset. Upon completion the results of the plot can be retrieved using the "Quality Control" link under the "Saved Analysis" menu.



Quality Control

This page allows you to view various quality control diagrams and metrics for the uploaded data. You may use the information gathered to identify arrays that may not be suitable for further use in the experiment.

Your currently selected experiment is '**ubertest**'.

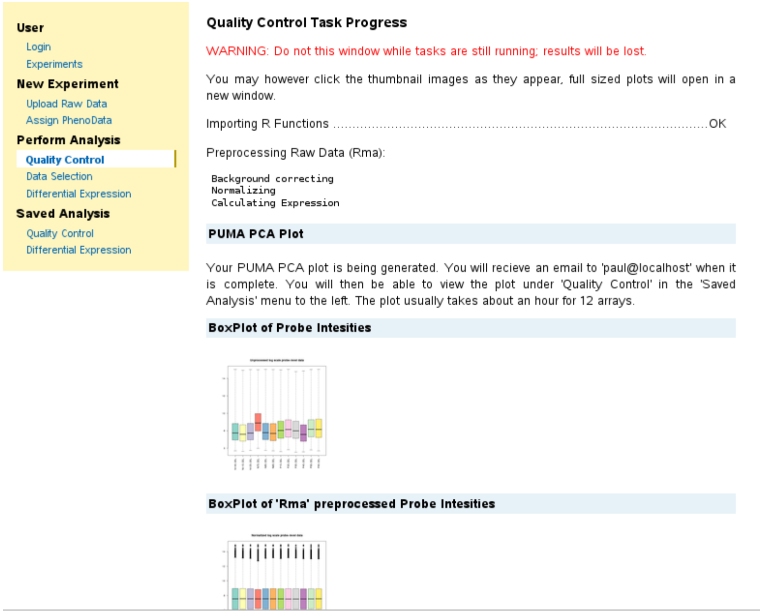
Output	Raw Data	Preprocessed Data
Boxplot	☑	☑
Histogram	☑	☑
PCA & Scree Plots		☑
PCA & Scree Plots (PUMA)		☑
RNA Degradation Plot	☑	
Simple		
RLE Pl		
NUSE		
Array Images	☑	
Affymetrix Quality Metrics	☑	

A SimpleAffy plot is a plot of various metrics recommended by AffyMetrix. To find out how to effectively read this plot, click HERE to read the users guide.

Analysis Name: Quality Control

Submit Query

Figure 3.7: The page displaying quality control options. Note tooltip displayed when hovering over "SimpleAffy Plot" option.



Quality Control Task Progress

WARNING: Do not this window while tasks are still running; results will be lost.

You may however click the thumbnail images as they appear, full sized plots will open in a new window.

Importing R Functions ....OK

Preprocessing Raw Data (Rma):

```
Background correcting
Normalizing
Calculating Expression
```

PUMA PCA Plot

Your PUMA PCA plot is being generated. You will recieve an email to 'paul@localhost' when it is complete. You will then be able to view the plot under 'Quality Control' in the 'Saved Analysis' menu to the left. The plot usually takes about an hour for 12 arrays.

BoxPlot of Probe Intesities

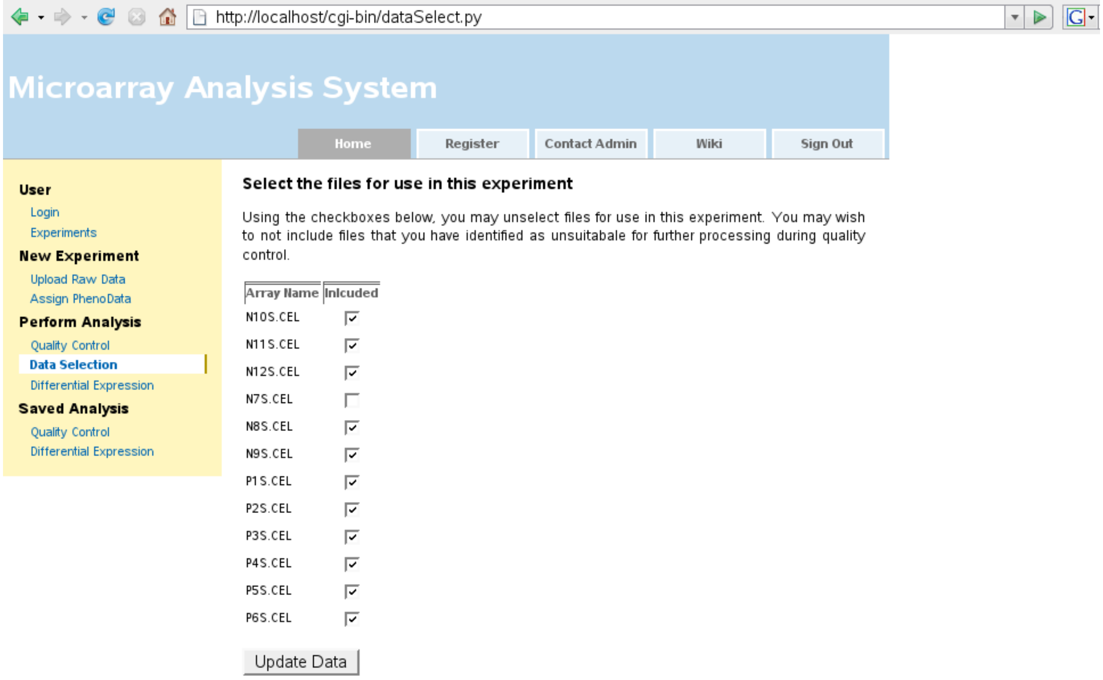
BoxPlot of 'Rma' preprocessed Probe Intesities

Figure 3.8: The page displaying quality control output (page heading removed from image).

3.6 Data Selection

The data selection phase may be required following quality control. There are certain situations where a user may decide that, for quality reasons, an array is not suitable for inclusion in further analysis. The data selection page allows a user to exclude an array from subsequent analysis without having to create a new experiment.

If for example we were to decide to exclude the array "NS7.CEL" of our bovine dataset, from differential expression analysis, we can do so using this page by simply unchecking here the array and pressing the "Update Data" button.



Select the files for use in this experiment

Using the checkboxes below, you may unselect files for use in this experiment. You may wish to not include files that you have identified as unsuitabale for further processing during quality control.

Array Name	Inlcuded
N10S.CEL	☑
N11S.CEL	☑
N12S.CEL	☑
N7S.CEL	☐
N8S.CEL	☑
N9S.CEL	☑
P1S.CEL	☑
P2S.CEL	☑
P3S.CEL	☑
P4S.CEL	☑
P5S.CEL	☑
P6S.CEL	☑

Figure 3.9: The data select page, shown excluding NS7.CEL from our bovine dataset.

3.7 Differential Expression

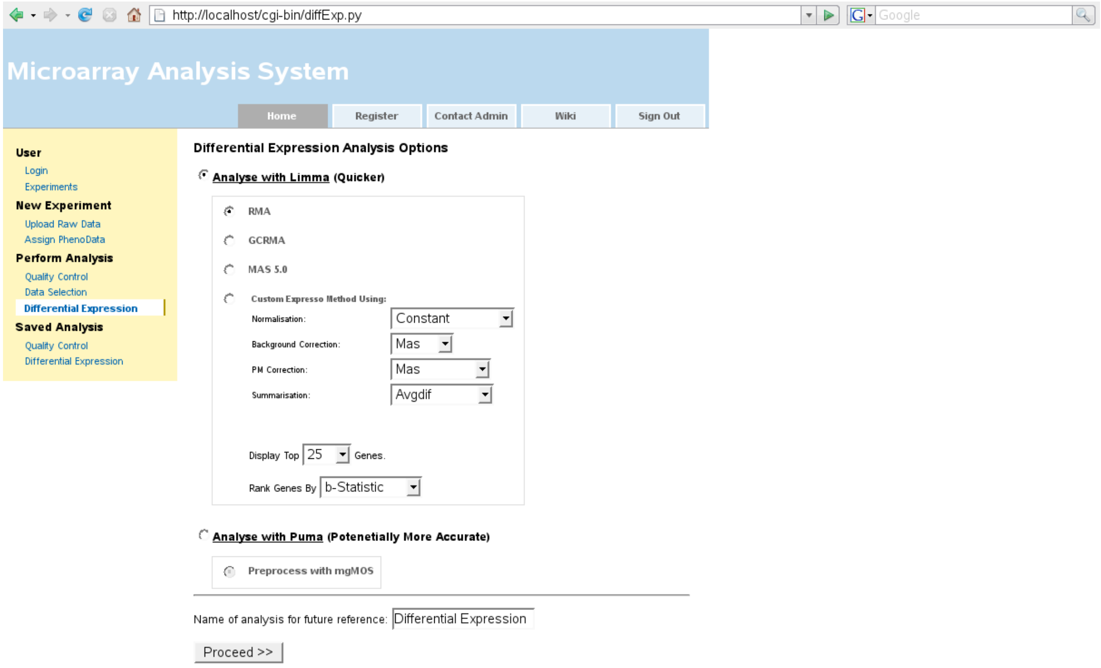
The next step in an analysis is to run differential expression. There are numerous different choices available at this stage and options differ significantly based on the type of dataset involved.

The figure below shows the initial options available if analysing an Affymetrix dataset. As you can see the user can select from the preprocessing and differential expression options that were outlined in chapter 2.

Once the user selects between the Limma and PUMA packages and selects a preprocessing method, they are required to decide on which contrasts they wish to test for differential expression. This is simple for our bovine dataset, as this is an experimental design of only one factor which has only two levels; so the only contrast available is between these two levels.

But now consider the 2x2 factorial estrogen dataset described previously in this chapter. In such a situation there are obviously a several contrasts available, some which may be worth examining and some which may not. Fig 5.11 on the next page shows which contrasts have been identified by the system from the phenotypic data that has been supplied.

Two contrasts which are obviously of interest are “Present +10 VS Absent +10” and “Present +48 VS Absent +48” which will tell us which genes are calcu-



Differential Expression Analysis Options

● Analyse with Limma (Quicker)

● RMA

○ GCRMA

○ MAS 5.0

○ Custom Expresso Method Using:

Normalisation: Constant

Background Correction: Mas

PM Correction: Mas

Summarisation: Avgdif

Display Top 25 Genes.

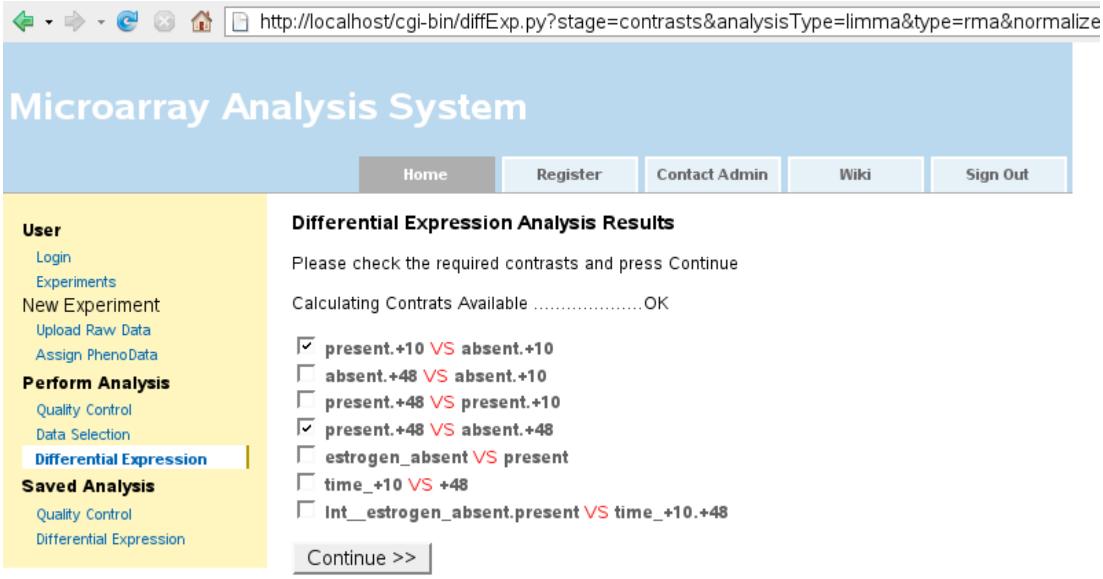
Rank Genes By b-Statistic

○ Analyse with Puma (Potenetially More Accurate)

● Preprocess with mgMOS

Name of analysis for future reference: Differential Expression

Figure 3.10: Differential expression options displayed for an Affymetrix dataset.



Differential Expression Analysis Results

Please check the required contrasts and press Continue

Calculating Contrats Available ....OK

☑ present.+10 VS absent.+10
□ absent.+48 VS absent.+10
□ present.+48 VS present.+10
☑ present.+48 VS absent.+48
□ estrogen_absent VS present
□ time_+10 VS +48
□ Int__estrogen_absent.present VS time_+10.+48

Continue >>

Figure 3.11: Contrasts available for the Estrogen Dataset.

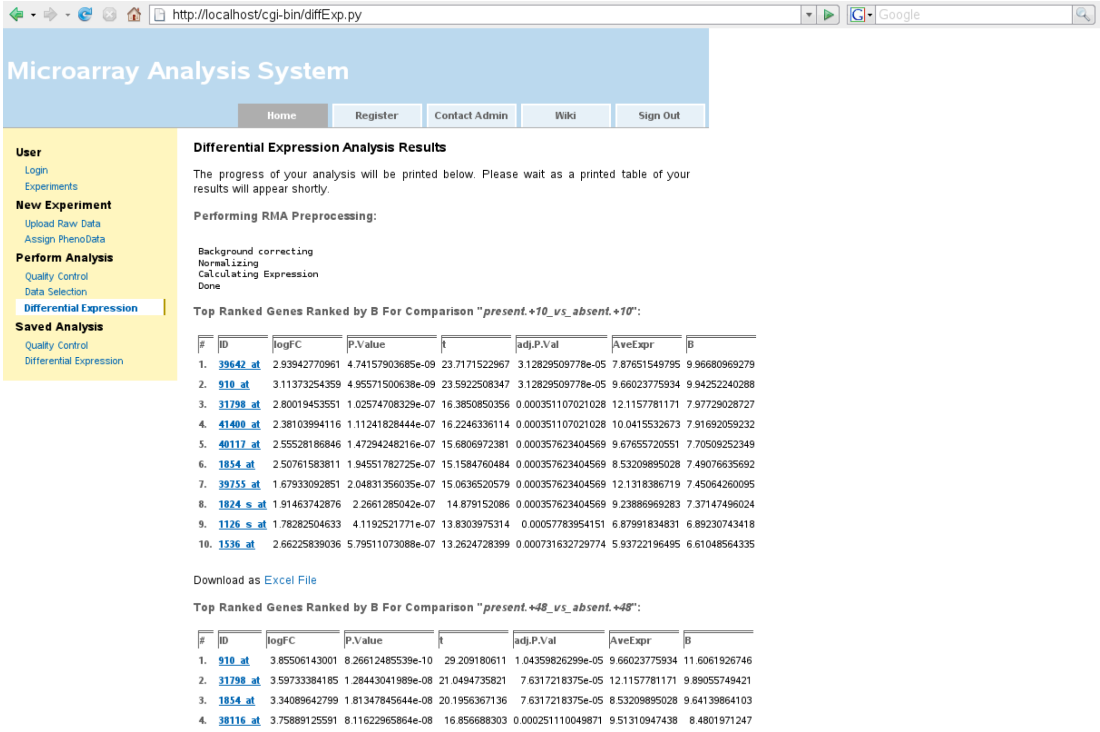
lated as being differentially expressed between the estrogen absent and estrogen present groups at 10 hours and subsequently at 48 hours.

After checking the contrasts of interest and submitting the page, differential expression analysis will begin.

Fig 3.12 on the top of the next page shows the output of differential expression analysis run for the Estrogen Dataset, where we have selected “Present +10 VS Absent +10” and “Present +48 VS Absent +48” as our contrasts of interest.

As you can see these results can also be downloaded as a .xls format spreadsheet file, which can be viewed in an application such as Microsoft Excel or OpenOffice.

If the user opts to use the PUMA method to assess differential expression, they must once again begin by specifying the contrasts of interest; once these are submitted a message is printed notifying the user that they will receive an email upon completion of differential expression analysis. This is because differential expression analysis using PUMA takes a long time, approximently 8 hours for our bovine dataset on a machine that boasts a 2GHz Intel ®Core^{TM}2 Duo CPU with 4 gigabytes of RAM. This can be reduced by parallelising the differential expression process across multiple cores of a single CPU, or across multiple processing cores of multiple machines, which is a topic that is discussed in the next chapter.



Differential Expression Analysis Results

The progress of your analysis will be printed below. Please wait as a printed table of your results will appear shortly.

Performing RMA Preprocessing:

```
Background correcting
Normalizing
Calculating Expression
Done
```

Top Ranked Genes Ranked by B For Comparison "*present.+10_vs_absent.+10*":

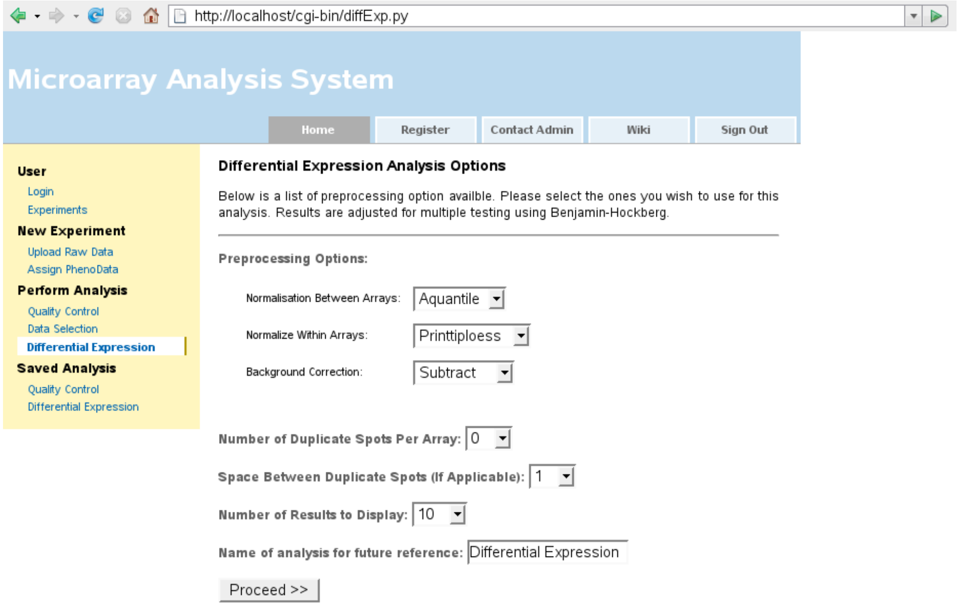
#	ID	logFC	P.Value	t	adj.P.Val	AveExpr	B
1.	39642_at	2.93942770961	4.74157903685e-09	23.7171522967	3.12829509778e-05	7.87651549795	9.96680969279
2.	910_at	3.11373254359	4.95571500638e-09	23.5922508347	3.12829509778e-05	9.66023775934	9.94252240288
3.	31798_at	2.80019453551	1.02574708329e-07	16.3850850356	0.000351107021028	12.1157781171	7.97729028727
4.	41400_at	2.38103994116	1.11241828444e-07	16.2246336114	0.000351107021028	10.0415532673	7.91692059232
5.	40117_at	2.55528186846	1.47294248216e-07	15.6806972381	0.000357623404569	9.67655720551	7.70509252349
6.	1854_at	2.50761583811	1.94551782725e-07	15.1584760484	0.000357623404569	8.53209895028	7.49076635692
7.	39755_at	1.67933092851	2.04831356035e-07	15.0636520579	0.000357623404569	12.1318386719	7.45064260095
8.	1824_s_at	1.91463742876	2.2661285042e-07	14.879152086	0.000357623404569	9.23886969283	7.37147496024
9.	1126_s_at	1.78282504633	4.1192521771e-07	13.8303975314	0.00057783954151	6.87991834831	6.89230743418
10.	1536_at	2.66225839036	5.79511073088e-07	13.2624728399	0.000731632729774	5.93722196495	6.61048564335

Download as Excel File

Top Ranked Genes Ranked by B For Comparison "*present.+48_vs_absent.+48*":

#	ID	logFC	P.Value	t	adj.P.Val	AveExpr	B
1.	910_at	3.85506143001	8.26612485539e-10	29.209180611	1.04359826299e-05	9.66023775934	11.6061926746
2.	31798_at	3.59733384185	1.28443041989e-08	21.0494735821	7.6317218375e-05	12.1157781171	9.89055749421
3.	1854_at	3.34089642799	1.81347845644e-08	20.1956367136	7.6317218375e-05	8.53209895028	9.64139864103
4.	38116_at	3.75889125591	8.11622965864e-08	16.856688303	0.000251110049871	9.51310947438	8.4801971247

Figure 3.12: List of differentially expressed genes in the Estrogen Dataset for the contrasts that we have selected.



Differential Expression Analysis Options

Below is a list of preprocessing option availble. Please select the ones you wish to use for this analysis. Results are adjusted for multiple testing using Benjamin-Hockberg.

Preprocessing Options:

Normalisation Between Arrays: Aquantile

Normalize Within Arrays: Printtiploess

Background Correction: Subtract

Number of Duplicate Spots Per Array: 0

Space Between Duplicate Spots (If Applicable): 1

Number of Results to Display: 10

Name of analysis for future reference: Differential Expression

Figure 3.13: Differential expression options for dual dye arrays

Differential expression analysis of dual and single dye data is similar to that of Affymetrix arrays, but because of the fact that the system currently only supports an experimental design where two levels of the same factor are compared, there is no need to specify contrasts, as there is only one contrast available. Preprocessing options also differ significantly, where the user has the choice of the options outlined in chapter 2.

Spots on dual and single dye arrays are sometimes duplicated one or more times within an array. This is to give a more reliable measure of the expression level of a gene represented by a spot. The system allows for this situation by allowing the user to specify the number of duplicate spots per array and the number of space between duplicates. Obviously this means that there must be the same number of duplicates for every spot and that the duplicates must be evenly spaced, but other than exceptional circumstances, this will always be the case. The level of correlation between these duplicate spots can then be factored into the linear model fit and subsequent differential expression analysis, hence giving a more accurate result. Fig. 3.13 on the bottom of the previous page shows the screen somebody beginning differential expression analysis of dual dye data will see.

As stated in chapter 3, gene annotation information can be dynamically downloaded by clicking the gene names in lists of differentially expressed genes. This information can then be interpreted by a biologist. The .RData file or the spreadsheet listing differentially expressed genes can be downloaded if subsequent analysis is to be pursued using an alternative platform.

3.8 Managing User Data

Experiments

The system makes a number of tools available to the user to manage data from previous experiments and analysis of experiments. The "Experiments" link in the menu on the left of the page will display a list of experiments that the user has previously uploaded and that are saved on the system, see Fig. 3.14. The option is available to remove any of these experiments from the system by checking them and clicking the delete key. The user may also select any previous experiments on the list for current use, which will allow them to either review previous analysis information or to perform new analysis. Clicking on any of the experiment names will bring the user to a page that shows various information about the dataset, such as the names of the files that were uploaded and data type.

Previous Quality Control Analysis

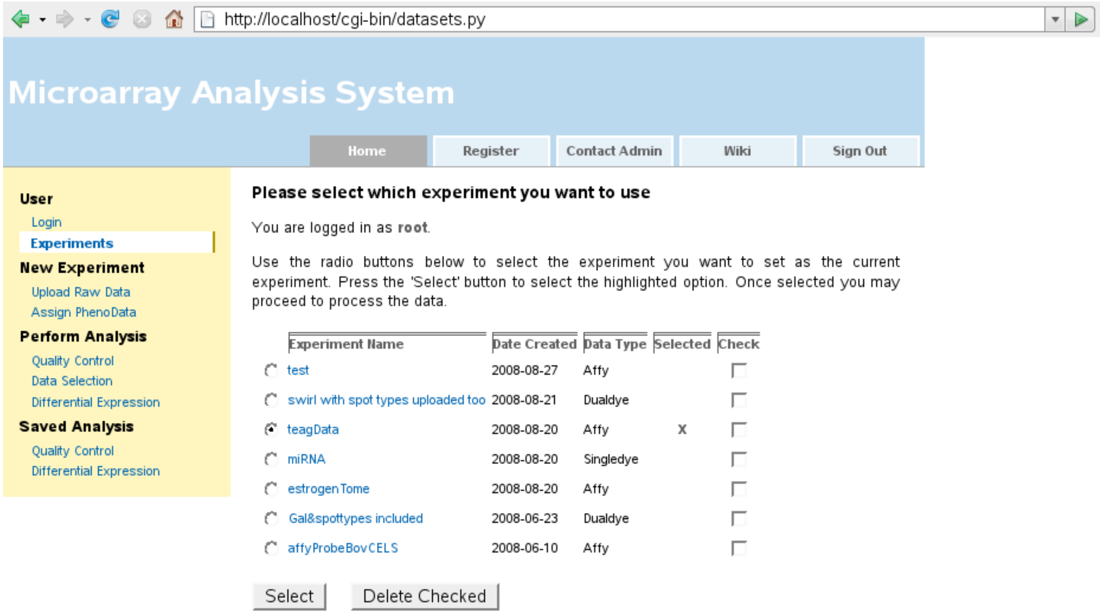
Every quality control analysis is automatically saved by system and can be reviewed at a later date by the user. This is done by clicking the “Quality Control” link under the “Saved Analysis” menu. Doing this will open a page with a list of all previous quality control analysis that have been completed for this experiment. This page also allows the user to delete these previous quality control analysis if desired. By clicking on any of the analysis the user is brought to a page that describes some of the conditions under which the analysis took place, such as which arrays were selected, which preprocessing methods were used and what phenotypic data was assigned.

If during quality analysis, the user had selected a PUMA PCA and scree plot, this is where they will find those plots upon their completion. As stated previously, the user will receive an email informing them of completion, the results can then be found under this menu (Fig. 3.15).

Previous Differential Expression Analysis

This is similar to the access of previous quality analysis. The user can delete previous analysis, access information on previous analysis, and view their results in full. The user also has the option of downloading the “.RData” file, which is saved when any given analysis is completed. This file contains all of the R objects that were created during the analysis. An advanced user may download this file and load it locally in R if they wish to pursue further analysis from the command line. Clicking the “More Info” link beside the link to download the .Rdata file will provide the user with a detailed description of what the objects saved in the file are and how they were created.

If the user has specified that analysis should be performed using PUMA, similarly to quality control analysis, this is where their results will appear upon completion.



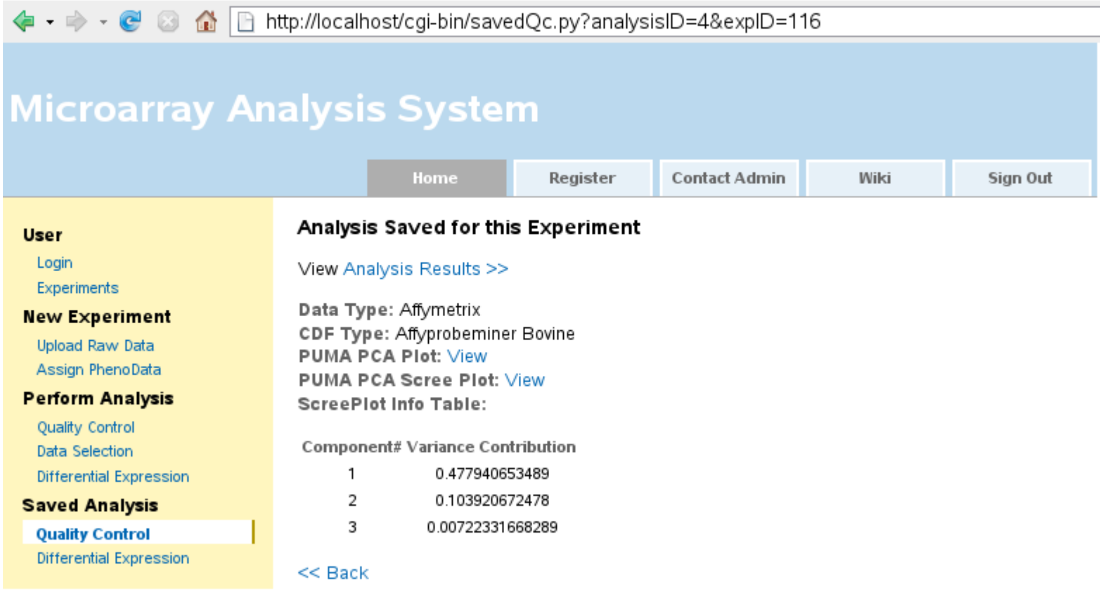
Please select which experiment you want to use

You are logged in as **root**.

Use the radio buttons below to select the experiment you want to set as the current experiment. Press the 'Select' button to select the highlighted option. Once selected you may proceed to process the data.

	Experiment Name	Date Created	Data Type	Selected	Check
○	test	2008-08-27	Affy		□
○	swirl with spot types uploaded too	2008-08-21	Dualdye		□
●	teagData	2008-08-20	Affy	X	□
○	miRNA	2008-08-20	Singledye		□
○	estrogenTome	2008-08-20	Affy		□
○	Gal&spottypes included	2008-06-23	Dualdye		□
○	affyProbeBovCELS	2008-06-10	Affy		□

Figure 3.14: A list of experiments that have been uploaded for this user. Any of these experiments may be clicked to review their content.



Analysis Saved for this Experiment

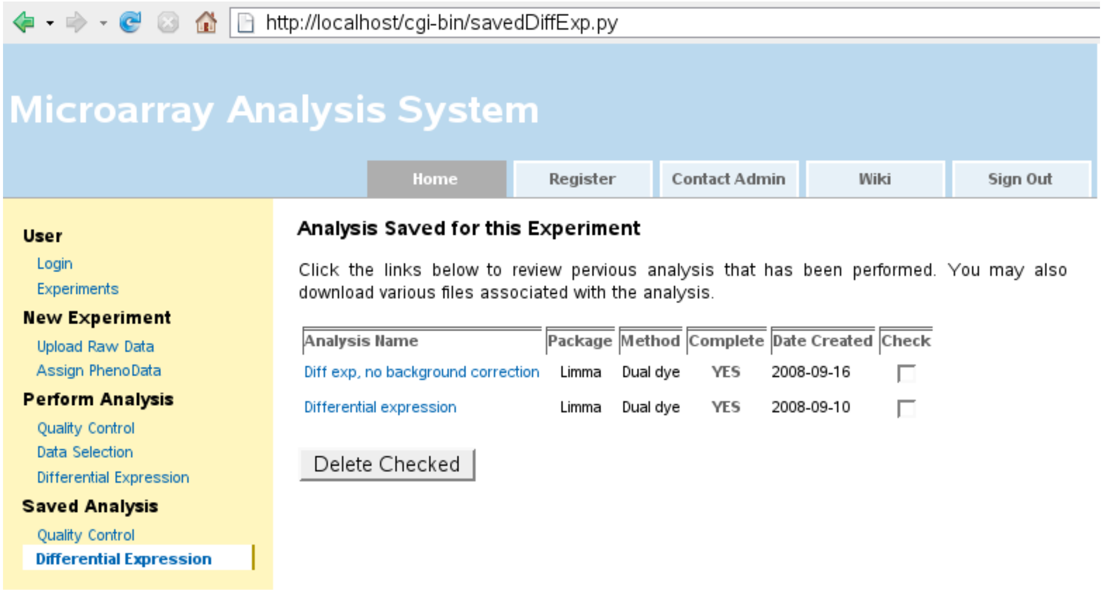
View Analysis Results >>

Data Type: Affymetrix
CDF Type: Affyprobeminer Bovine
PUMA PCA Plot: View
PUMA PCA Scree Plot: View
ScreePlot Info Table:

Component#	Variance Contribution
1	0.477940653489
2	0.103920672478
3	0.00722331668289

<< Back

Figure 3.15: Saved quality control page. Note how PUMA PCA and scree plots have been generated and are available here.

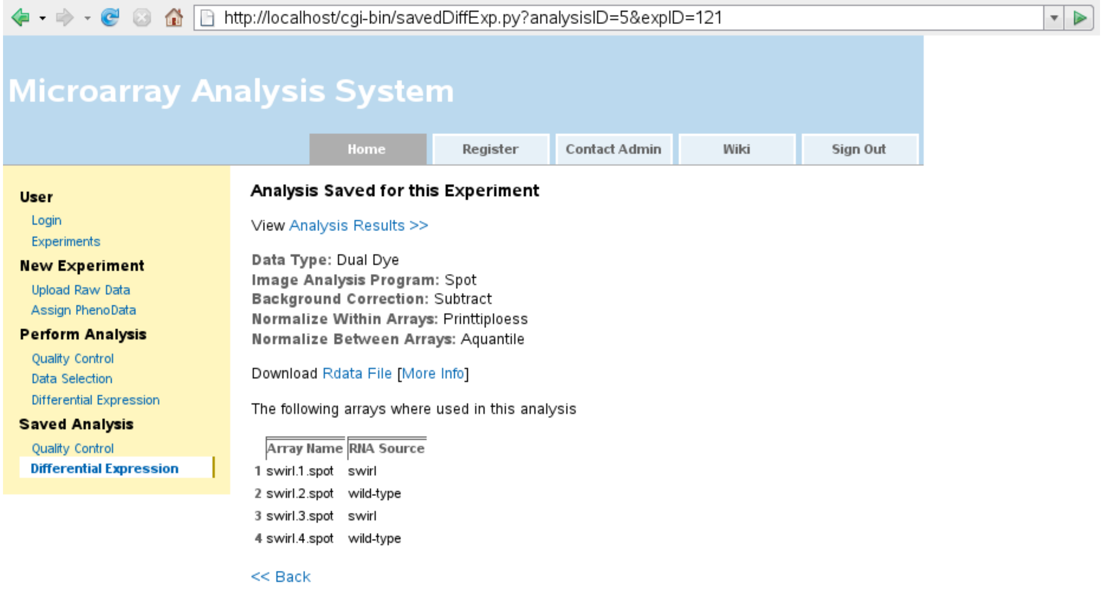


Analysis Saved for this Experiment

Click the links below to review pervious analysis that has been performed. You may also download various files associated with the analysis.

Analysis Name	Package	Method	Complete	Date Created	Check
Diff exp, no background correction	Limma	Dual dye	YES	2008-09-16	□
Differential expression	Limma	Dual dye	YES	2008-09-10	□

Figure 3.16: Page showing differential expression analysis that has been performed for the Swirl Dataset.



Analysis Saved for this Experiment

View Analysis Results >>

Data Type: Dual Dye
Image Analysis Program: Spot
Background Correction: Subtract
Normalize Within Arrays: Printtiploess
Normalize Between Arrays: Aquantile

Download Rdata File [More Info]

The following arrays where used in this analysis

	Array Name	RNA Source
1	swirl.1.spot	swirl
2	swirl.2.spot	wild-type
3	swirl.3.spot	swirl
4	swirl.4.spot	wild-type

<< Back

Figure 3.17: Details displayed after a particular analysis is clicked

Chapter 4

Installing the System

4.1 Using the BioconductorBuntu Installation CD

The BioconductorBuntu Linux distribution was created with the intention of greatly simplifying the process of getting a microarray analysis web server up and running. Installation is performed by simply downloading the systems ISO file, which is burned to a CD. The user must then boot from this CD, whereby an extremely simple and user friendly process guides them through the installation.

For testing purposes the system can also be installed on a virtual machine, such as VM Ware Server.

The steps taken for a new install are precicely as follows:

- Download the BioconductorBuntu ISO file and burn to a blank CD.
- Insert new CD which you have created from the downloaded ISO file, restart the computer and boot from the CD.
- Select the “Start Custom Live CD in Graphical Mode” option.
- When the system starts, double click the “Install” icon on the desktop.
- Follow the instructions, creating user account and password etc.
- Once the installation is finished, restart and boot into Linux. You should have removed the CD.
- Login using the username and password you created during installation.
- To access the microarray analysis system, open the Firefox web-browser and navigate to “http://localhost/cgi-bin/login.py”

- You may create a new account or login to an example account that has been created which has username “bioconductor” and password “microarray”
- To access the system remotely, replace the word “localhost” in the URL above with the IP address or hostname of the machine on which the system is now installed.

4.2 Installing individual System Components Manually

A user with a reasonable knowlege of Linux should be able to install this systems individual components on an existing Linux server. This process has been successfully complete on Ubuntu & OpenSUSE. The following steps need to be taken.

If possible, for the sake of simplicity, you should install the components using the systems package manager. The instructions below are targeted at installing on a Ubuntu Linux system but can be adapted for any system.

- If not already done, install MySQL using package manager:

  ```
  ~$ sudo apt-get install mysql-server
  ~$ sudo apt-get install mysql-client-5.0 libmysqlclient15-dev
  ```

- If not already done, install python using package manager:

  ```
  ~$ sudo apt-get install python
  ```

- Install mail server:

  ```
  ~$ sudo tasksel install mail-server
  ```

- Build R from source, as shared library. Depending on how your system is set up you may have to resolve dependencies like, libx11-dev,libxt-dev r-base-dev, gfortran and python-dev. These packages will vary from system to system, but the following lines will compile and install R on any Linux system where 'R.tar.gz' is the name of the R source file you have downloaded:

  ```
  ~$ sudo tar -zxvf R.tar.gz
  ~$ sudo ./configure --enable-R-shlib
  ~$ sudo make
  ~$ sudo make install
  ```

- Install bioconductor:

  ```
  ~$ sudo R
  > source("http://bioconductor.org/biocLite.R")
  > biocLite()
  ```

- Install Rpy:

  Configure the path to the R library. You have several ways to do this (substitute RHOME with the path where R is installed, usually /usr/local/lib/R):

  Make a link to RHOME/bin/libR.so in /usr/local/lib or /usr/lib, then run:

  ```
  ~$ sudo ldconfig
  ```

  Or, edit the file /etc/ld.so.conf and add the following line:

  ```
  RHOME/bin
  ```

  and then, run 'ldconfig' as above.

  On 64 bit, this seems to be RHOME/lib rather than RHOME/bin. So the line added to /etc/ld.so.conf is "/usr/lib64/R/lib/"

- Install imagemagick from package manager:

  ```
  ~$ sudo apt-get install imagemagick
  ```

- Setup and configure Apache. On Ubuntu this command will setup a LAMP server:

  ```
  ~$ sudo tasksel install lamp-server
  ```

- Install MySqlDb in Python:

  ```
  ~$ sudo apt-get install python-mysqldb
  ```

- All bioconductor packages may not be installed, for example the bovine cdf file. Puma, biomaRt, affycoretools, simpleaffy, statmod etc.

  ```
  ~$ sudo R
  > source("http://bioconductor.org/biocLite.R")
  > biocLite("Puma")
  > biocLite("biomaRt")
  > biocLite("affycoretools")
  > biocLite("simpleaffy")
  > biocLite("statmod")
  ```

- To get biomaRt working you need curl developement libraries and XML libraries installed:

  ```
  ~$ sudo apt-get install libcurl3-dev
  ~$ sudo apt-get install libxml2-dev
  ```

- You need to upload the bovinecdf.qcdef file to the right directory, so that the bovine file can be used with simpleaffy.

- You now need to put the relevant files in the relevant directories. On Ubuntu cgi-bin is "/usr/lib/cgi-bin" and htdocs is "/var/www/". You need to create a directory somewhere for the R scripts and put them in there. You now need to edit the file MicroSiteTemplate.py between lines 30 and 43. These lines contain constants that point to the various directories on the system in question, for example the cgi-bin and htdocs directory. You should know what these are on your system.

Bibliography

[1] Affymetrix. *Guide to Probe Logarithmic Intensity Error (PLIER) Estimation.* Affymetrix.

[2] Affymetrix. Statistical algorithms description document., 2002.

[3] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *, Journal of the Royal Statistical Society*, 57:125–133, 1995.

[4] J Brettschneider K Simpson L Cope RA Irizarry TP Speed BM Bolstad, F Collin. *Quality Assessment of Affymetrix GeneChip Data.* Springer, 2003.

[5] B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, Jan 2003.

[6] Manhong Dai, Pinglang Wang, Andrew D Boyd, Georgi Kostov, Brian Athey, Edward G Jones, William E Bunney, Richard M Myers, Terry P Speed, Huda Akil, Stanley J Watson, and Fan Meng. Evolving gene/transcript definitions significantly alter the interpretation of genechip data. *Nucleic Acids Res*, 33(20):e175, 2005.

[7] Laurent Gautier, Leslie Cope, Benjamin M Bolstad, and Rafael A Irizarry. affy–analysis of affymetrix genechip data at the probe level. *Bioinformatics*, 20(3):307–315, Feb 2004.

[8] Peter Huber. *Robust Statistics.* Wiley, New York, 1981.

[9] Wolfgang Huber, Anja von Heydebreck, Holger SÃ¼ltmann, Annemarie Poustka, and Martin Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18 Suppl 1:S96–104, 2002.

[10] RA Irizarry, Laurent Gautier, and L. Cope. *The Analysis of Gene Expression Data.* Springer, 2003.

[11] Rafael A Irizarry, Benjamin M Bolstad, Francois Collin, Leslie M Cope, Bridget Hobbs, and Terence P Speed. Summaries of affymetrix genechip probe level data. *Nucleic Acids Res*, 31(4):e15, Feb 2003.

[12] D. A. Lashkari, J. L. DeRisi, J. H. McCusker, A. F. Namath, C. Gentile, S. Y. Hwang, P. O. Brown, and R. W. Davis. Yeast microarrays for genome wide parallel genetic and gene expression analysis. *Proc Natl Acad Sci U S A*, 94(24):13057–13062, Nov 1997.

[13] R. J. Lipshutz, S. P. Fodor, T. R. Gingeras, and D. J. Lockhart. High density synthetic oligonucleotide arrays. *Nat Genet*, 21(1 Suppl):20–24, Jan 1999.

[14] Hongfang Liu, Barry R Zeeberg, Gang Qu, A. Gunes Koru, Alessandro Ferrucci, Ari Kahn, Michael C Ryan, Antej Nuhanovic, Peter J Munson, William C Reinhold, David W Kane, and John N Weinstein. Affyprobeminer: a web resource for computing or retrieving accurately redefined affymetrix probe sets. *Bioinformatics*, 23(18):2385–2390, Sep 2007.

[15] Xuejun Liu, Marta Milo, Neil D Lawrence, and Magnus Rattray. A tractable probabilistic model for affymetrix probe-level analysis across multiple chips. *Bioinformatics*, 21(18):3637–3644, Sep 2005.

[16] Xuejun Liu, Marta Milo, Neil D Lawrence, and Magnus Rattray. Probe-level measurement error improves accuracy in detecting differential gene expression. *Bioinformatics*, 22(17):2107–2113, Sep 2006.

[17] J. Nielsen. *Ten Usability Heuristics.* http://www.useit.com/papers/heuristic/, 2005.

[18] Helen Pearson. Genetics: what is a gene? *Nature*, 441(7092):398–401, May 2006.

[19] R.D. Pearson, X. Lui, and M.Rattray. puma: a bioconductor package for propagating uncertainty in microarray analysis. 2007.

[20] Matthew E Ritchie, Jeremy Silver, Alicia Oshlack, Melissa Holmes, Dileepa Diyagama, Andrew Holloway, and Gordon K Smyth. A comparison of background correction methods for two-colour microarrays. *Bioinformatics*, 23(20):2700–2707, Oct 2007.

[21] David M Rocke and Blythe Durbin. Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics*, 19(8):966–972, May 2003.

[22] G. Ruvkun. Molecular biology. glimpses of a tiny rna world. *Science*, 294(5543):797–799, Oct 2001.

[23] Guido Sanguinetti, Marta Milo, Magnus Rattray, and Neil D Lawrence. Accounting for probe-level noise in principal component analysis of microarray data. *Bioinformatics*, 21(19):3748–3754, Oct 2005.

[24] D. Scholtens and A. von Heydebreck. *Analysis of Differential Gene Expression Studies*. Springer London, 2005.

[25] Holger Schwender and Anton Belousov. Comparison of preprocessing methods for affymetrix microarrays. *A Magazine of the American Statistical Association*, 19(3):16, Summer 2006.

[26] Gordon K Smyth. Linear models and empirical bayes methods for assessing differential expression of microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1):Article 3, 2004.

[27] Gordon K. Smyth and Terry Speed. Normalization of cdna microarray data. *Methods*, 31:265–273, 2003.

[28] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, September 2003.

[29] Claire Wilson, Stuart D Pepper, and Crispin J Milller. Qc and affymetrix data.

[30] Z. J. Wu, R. A. Irizarry, R. Gentleman, F. Martinez-Murillo, and F. Spencer. A model-based background adjustment for oligonucleotide expression arrays. *Journal Of The American Statistical Association*, 99(468):909–917, December 2004.

[31] Yong You, Bernardo G Moreira, Mark A Behlke, and Richard Owczarzy. Design of lna probes that improve mismatch discrimination. *Nucleic Acids Res*, 34(8):e60, 2006.