



LUDWIG-MAXIMILANS-UNIVERSITÄT
TECHNISCHE UNIVERSITÄT MÜNCHEN



**Fakultät für Biologie
Department Biologie II
Lehrstuhl für Evolutionsbiologie
Evolutionary and Functional Genomics**

Diplomarbeit
in Bioinformatik

**Developing tools for the analysis of
microarray expression data**

Annahita Oswald

Aufgabensteller:
Betreuer:
Abgabedatum:

Prof. Dr. John Parsch
Prof. Dr. John Parsch
16. Juli 2007

Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

16. Juli 2007

Annahita Oswald

Abstract

This thesis describes a computational toolbox, called MuMAT (Munich Microarray Analysis Tool), that provides several tools for the analysis of microarray expression data. As processing information retrieved from high-throughput technologies like microarrays is a very complex and time consuming task, tools that analyze the data in an automated way are indispensable. In the course of this thesis new tools were added to MuMAT enabling the user to analyze the data in various ways. One of the new functions, called “BAGEL extractor”, splits the mass of data into manageable sub-data-sets which can be further manipulated. The “BAGEL summarizer” provides statistics that give information about the number and magnitude of expression differences between two or more biological samples, as well as the overall statistical power of the experiment.

Clustering biological samples of an experiment based on their gene expression patterns can be done using the “expression states” function in MuMAT that generates output files which can be used to construct trees using programs like PAUP* or Felsenstein`s PHYLIP.

MuMAT is available internally at the LMU Biozentrum (<http://10.153.66.19/MuMat>).

A downstream analysis presenting the variation in gene expression of Dutch and Zimbabwean populations of *Drosophila melanogaster* is performed as an example. Eight lines per population were used for the study. It is shown that the experiment has high statistical power in detecting expression differences between samples. Differential expression patterns between different lines as well as between the two populations are highlighted and a parsimony tree was constructed using PAUP* which clusters the strains on the basis of genes expression similarities.

Zusammenfassung

Diese Arbeit beschreibt MuMAT, das mehrere Programme zur Analyse von Microarray Expressionsdaten bereitstellt.

Da die Verarbeitung von Informationen die von high-throughput Methoden, wie der Microarray Technologie stammen, sehr komplex und zeitaufwändig ist, sind Programme die diese Analysen automatisieren unabdingbar. Im Rahmen dieser Arbeit wurden zu MuMat weitere Programme hinzugefügt, die es dem Benutzer ermöglichen die erlangten Daten auf verschiedene Weise zu beleuchten.

Die Methode „BAGEL extractor“, zerteilt die Menge an Daten in kleinere Datensätze, die dann weiter bearbeitet werden können. „BAGEL summarizer“ stellt mehrere Statistiken bereit, die über Anzahl und Größe von Expressionsunterschieden zweier oder mehrerer Samples und über die Mächtigkeit des Experiments Expressionsunterschiede zu erkennen informieren.

Biologische Samples eines Experiments können mit Hilfe der Funktion „expression states“ in MuMAT aufgrund ähnlicher Genexpression gruppiert werden. Diese stellt Dateien für Programme wie PAUP* oder PHYLIP von Felsenstein bereit, die daraus dann phylogenetische Bäume erstellen.

MuMAT ist innerhalb des LMU Biozentrums erreichbar (<http://10.153.66.19/MuMat>).

Zusätzlich wird eine Auswertung der Expressionsdaten durchgeführt, die die unterschiedliche Genexpression von holländischen und simbabwischen Populationen von *Drosophila Melanogaster* aufzeigt. Für das Experiment wurden acht Stämme aus jeder Population herangezogen. Es wird gezeigt, dass der Aufbau des Experiments auch sehr feine Expressionsunterschiede erkennen lässt. Unterschiedliche Genexpression sowohl zwischen den verschiedenen Stämmen, als auch zwischen den beiden Populationen werden aufgezeigt und ein Parsimony-Baum wird mittels PAUP* erstellt, der die Stämme basierend auf Ähnlichkeiten in der Genexpression gruppiert.

Table of Contents

1. Introduction	11
1.1. Motivation	13
1.2. Drosophila melanogaster as model organism.....	13
1.3. Microarrays.....	15
1.4. Pre-processing, Background Correction and Normalization	17
2. Materials and Methods	19
2.1. Drosophila melanogaster lines	19
2.2. Experimental design	19
2.3. BAGEL.....	21
2.4. BAGEL randomizer.....	22
2.5. BAGEL summarizer	22
2.5.1. Significant genes.....	22
2.5.2. FDR for each pairwise comparison	23
2.5.3. Number of up or down regulated genes	23
2.5.4. Up/down ratios for each pairwise comparison	24
2.5.5. List of significant differences per gene	24
2.5.6. Fold change.....	24
2.6. Discrete expression states.....	25
2.6.1. Rank Neighbor.....	25
2.6.2. Rank Reference.....	26
2.7. PAUP*	27

2.7.1. Parsimony Tree.....	27
3. MuMAT.....	30
3.1. BAGEL randomizer.....	30
3.2. BAGEL extractor.....	32
3.3. BAGEL summarizer.....	36
3.4. Discrete expression states.....	41
4. Results.....	46
4.1. Comparison of individual lines.....	46
4.2. Inter-population comparison.....	55
4.3. Expression tree.....	57
5. Conclusion.....	63
Appendix.....	65
A) List of Figures.....	65
B) List of Tables.....	67
C) Content of CD.....	68
Acknowledgements.....	69
Bibliography.....	71

1. Introduction

Microarrays, as high-throughput technologies, inform on the measurement of mRNA levels in particular cells or tissues for many genes at once. They allow the detection of differentially expressed genes in different cell stages of an organism, different individuals, strains or whole populations. The mass of data resulting from such experiments has underscored the importance of computational analysis as a key link between data generation and the formulation of new hypotheses. There are many tools available for the analysis of microarray data. Approaches like clustering or phylogenetic systematics are important steps in the analysis of microarray data. In this thesis, I will point out the relevance of such tools and focus on a computational toolbox, called MuMAT (Munich Microarray Analysis Tool), that combines several methods for the statistical analysis of microarray expression data in one interface. I will highlight the steps from the experimental part to the point of the analysis of the retrieved data and conclude with some challenges that can be added to this toolbox to expand the opportunities for analysis.

Figure 1.1 shows the workflow of a comparative analysis between two different lines of *Drosophila melanogaster* which I focus on in this thesis.

The scheme shows that first the mRNAs out of the samples are extracted and an expression profile of each of the sample is generated. Two different populations of *Drosophila melanogaster*, the Dutch and Zimbabwean population, were taken for the experiments. An analysis of expression differences within the populations as well as between the populations is performed. In order to estimate expression differences within the populations, eight different strains of each population are taken.

Second, the profiles are prepared for further analyses, that is they were scanned, and a so-called Spot Finding is performed by a special software. In the end this software produces statistics which give information about the foreground and background intensities of every single spot. In a next step a quality check has to be performed and the statistics have to be normalized because of variances in the sample preparation, labeling or signal measurement. After the normalization of all arrays the gene expression levels are determined together with some other information by using a stand-alone software called BAGEL.

The output of the BAGEL run is then analyzed using MuMAT.

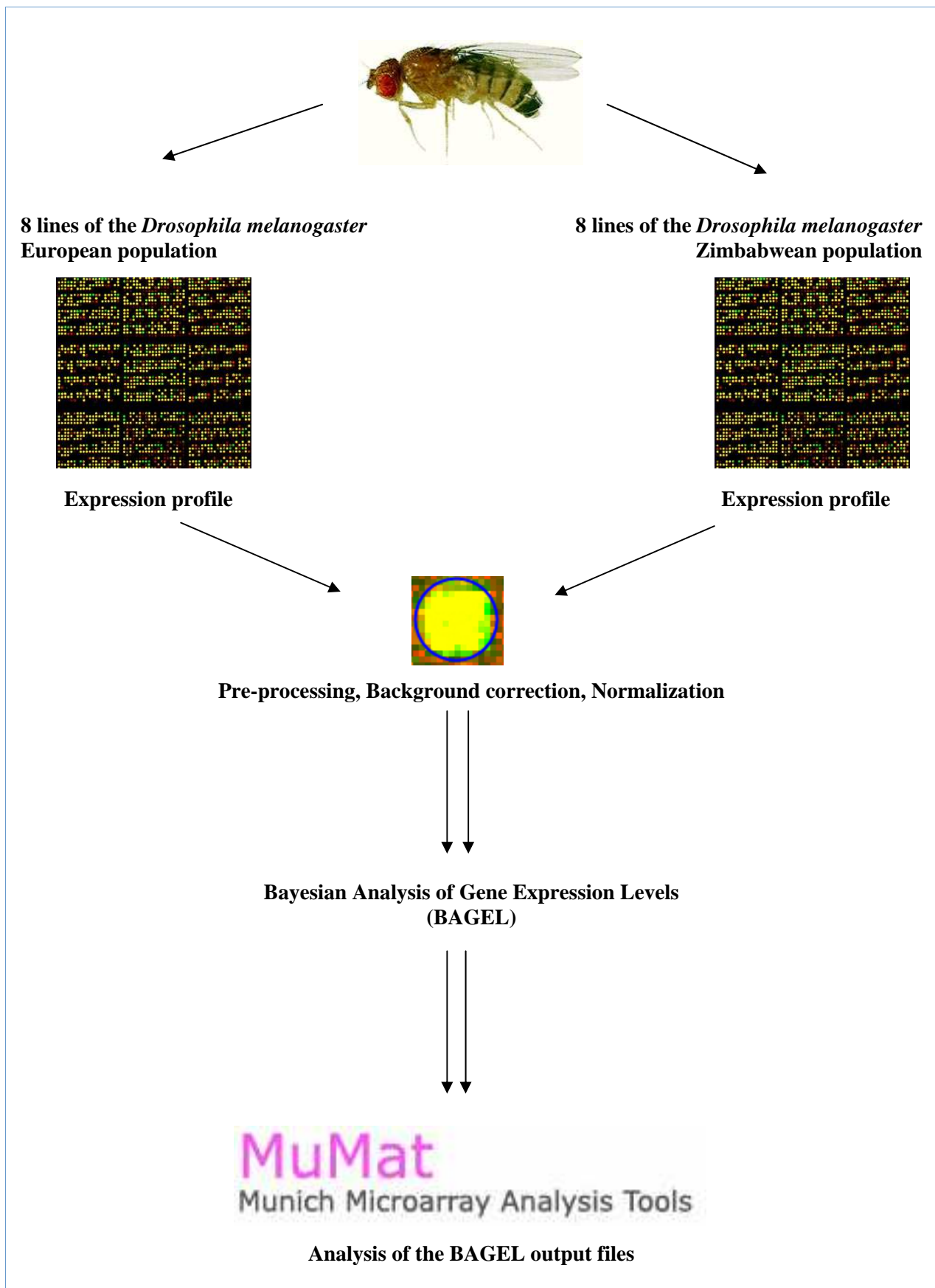


Figure 1.1 Workflow of this thesis

1.1. Motivation

As more DNA sequences became available, methods were employed to analyze these genes in various ways. Technologies like microarray analysis arose. This type of genome analysis is used to study regulation or sequence variation of thousand of genes or synthesized proteins at the same time. The major use of microarray technology has been to monitor expression of genes simultaneously in a given biological sample or time point in an experiment. Microarrays facilitate conclusions about which genes are being expressed at a higher or lower level or at the same level in different samples. It gives useful biological information, e.g. which genes are induced or repressed in a phase of the cell cycle or which genes are differentially expressed in different strains.

Considerable data is collected from these analyses. Keeping track of so much data is beyond the capability of most laboratories.

Analyzing the mass of data that is produced by technologies like microarray experiments manually, is a time-consuming, error-prone, and resource-wasteful process.

Therefore the development of computational methods that automate these processes is needed. Bioinformatic tools support a broad spectrum of research that includes determining the biological significance of the data, provide the expertise to organise it, and provide methods to mine the data for new information (Mount 2004).

This thesis describes tools that enable statistical analysis of microarray expression data in an automated way. These tools provide statistical methods and programs that help the user to manipulate the data that come along with microarray experiments. All these methods are provided in a user interface called MuMAT, Munich Microarray Analysis Tool that is accessible within the LMU-Biozentrum (<http://10.153.66.19/MuMAT/>).

1.2. *Drosophila melanogaster* as model organism

Drosophila melanogaster belongs to a closely related group of eight species collectively known as the *melanogaster* subgroup (Caccone 1996). The phylogenetic relationships of this group are shown in Figure 1.2. All are native to sub-Saharan Africa and islands off the east coast of Africa. The existence of close relatives facilitates comparative work, so the group has become a paradigm for speciation studies.

Drosophila melanogaster is a fruit fly, a little insect about 3mm long. It is one of the most valuable organisms in biological research, particularly in genetics and developmental biology. Since its initial use in the development of the field of genetics, *Drosophila melanogaster* has held a central position in biological research and serves as a model system for the investigation of many developmental and cellular processes common to higher eukaryotes, including humans (Adams, Celniker et al. 2000).

The reason why people work on it is that so much is already known about it and it is easy to handle and well-understood. It's a small animal, with a short life cycle of just two weeks, it is cheap and it's easy to keep large numbers. The large populations make statistical analysis easy and reliable.

The size of the genome is approximately 165 million bases and contains about 14,000 genes. The genome was (almost) completely sequenced in 2000 which enabled researchers to make different forms of analyses on the genome.

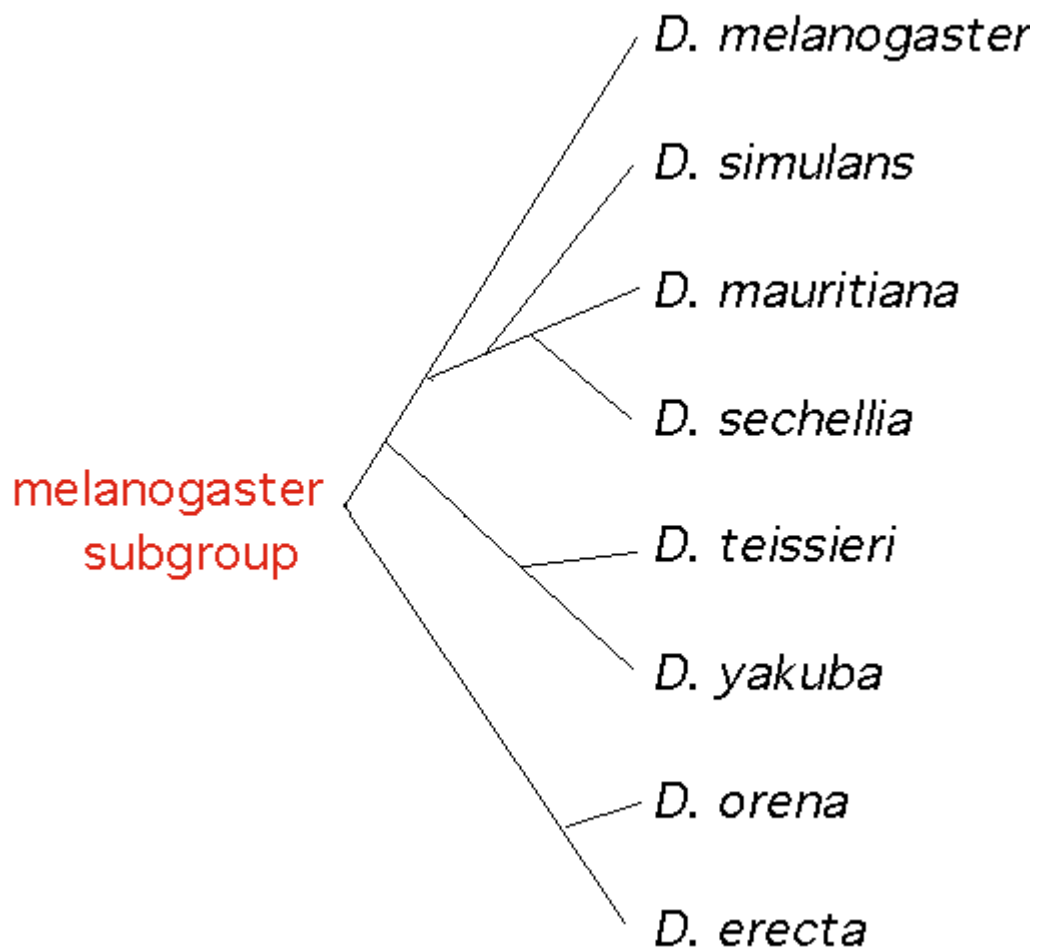


Figure 1.2 Phylogeny of the *Drosophila melanogaster* subgroup (Powell 1997)

1.3. Microarrays

Although all of the cells in the human body contain the same genetic material, the same genes are not active in all of those cells. Studying which genes are active and which are inactive in different kinds of cells or phases of the cell cycle helps scientists to understand more about how these cells function and about what happens when the genes in a cell don't function properly.

In the past scientists have only been able to make such genetic analyses on a few genes at once, but with the development of DNA microarray technology, however, it is now possible to examine thousands of genes at the same time which enables to determine the complex relationships between individual genes.

There are two types of microarrays, the cDNA or spotted arrays and the high-density oligonucleotide arrays.

On cDNA arrays each spot contains a cDNA clone from a known gene. Since cDNA clones are much longer than oligonucleotides, a successful hybridization with a clone is an almost certain match for the gene. This facilitates quickly a profile of expression levels of known genes. With cDNA arrays it is possible to compare the expression levels of two different samples, one test and one reference, which are differently fluorescently labeled with green (Cy3) or red (Cy5) dyes, on the same chip. Absolute levels of gene expression cannot be determined with this type of arrays but just relative ones.

In contrast to the cDNA arrays the oligonucleotide arrays contain about 20 short oligos approximately 25 base pairs in length for each gene plus the same amount of mismatch controls with single nucleotide mismatch in the center. The expression level is measured as the intensity difference between match and mismatch over all segments of a gene. In this method, mRNA from a single biological sample is hybridized to the oligonucleotides of the array, this implies that two separate arrays are needed for the comparison of two biological samples, whereas only one slide is needed with the cDNA array method. One big advantage is that these microarrays give estimations of the absolute value of gene expression.

Here we only focus on the spotted arrays. Figure 1.3 shows the major steps in a microarray experiment.

- ***DNA microarray making***

DNA fragments amplified by PCR technique are spotted on a poly-lysine coated support typically a glass slide, a quartz wafer, or a nylon membrane with a robotic arrayer. The poly-lysine that is not fixed to DNA is blocked in order to avoid target binding. In a next step the DNA is denatured to obtain a single strand DNA on the microarray. This will allow the probe to bind to the complementary strand from the target. In our case a genome-wide *Drosophila melanogaster* microarray from the Drosophila Genomics Resource Center (DGRC) was used. The microarray contains 13921 PCR products representing 11895 unique genes. Not all genes are present on the microarray, but of some genes more than one replicate is represented.

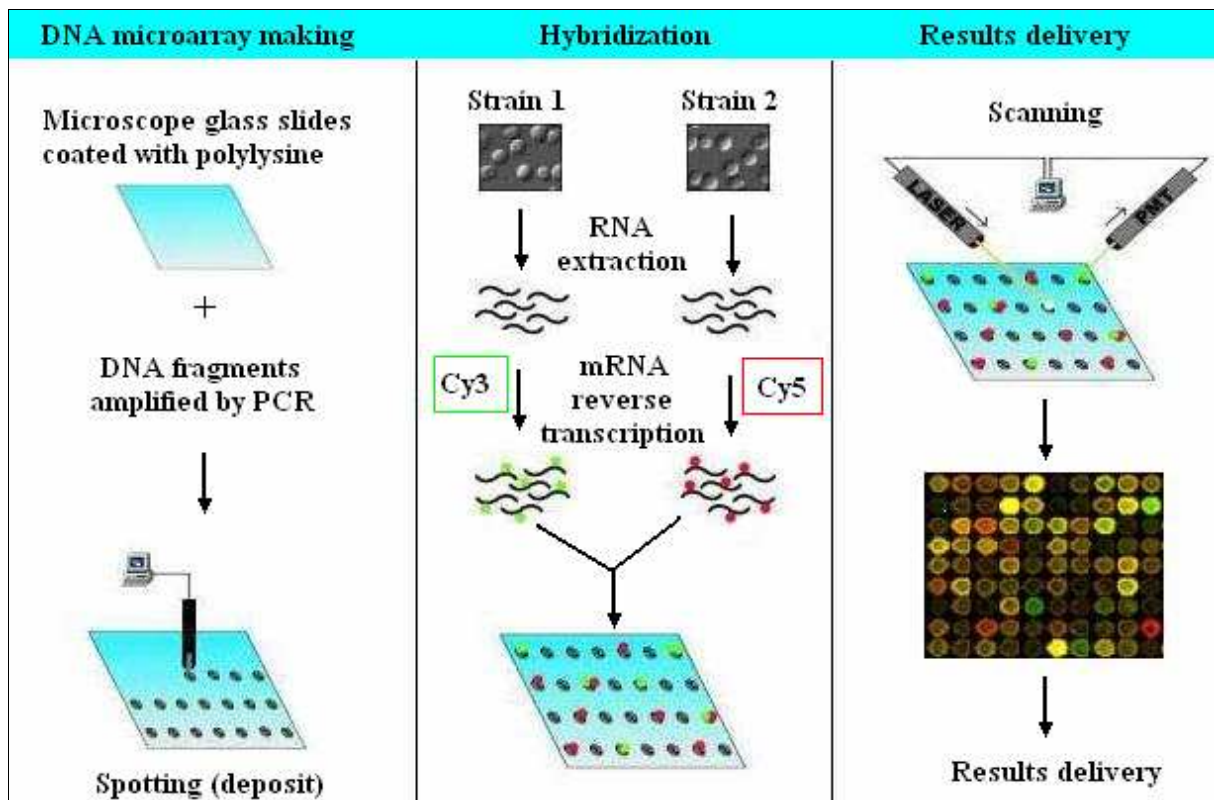


Figure 1.3 Major steps in DNA microarray experiments
(<http://www.transcriptome.ens.fr/sgdb/>)

- **Hybridization**

From two different samples to be compared messenger RNA is obtained. The mRNAs are then transformed in cDNA by reverse transcription and are labeled with two different fluorophores, green (Cy3) and red (Cy5) dyes, respectively. A mixture of the labeled cDNAs is then hybridized to the slide. The chip is incubated one night at 60 degrees. The fluorescent DNA will then hybridize on the spotted ones, because at this temperature complementary DNA strands that match together create a double strand DNA.

- **Results delivery**

After the hybridization the slide is scanned with a microscope to measure the amount of label hybridized to each spot. The ratios of the labels give information about the ratio of mRNA levels in the original samples. The ratio is indicated by a color (Mount 2004). Figure 1.4 shows an example of a cDNA array. The colors represent the ratios of the Cy3 and Cy5 labels at each spot. The parts of the array with no color represent the spots where no labeled sample DNA is bound. The red spots show the spots where red labeled DNA is bound whereas the green spots indicate bound green labeled DNA. Yellow spots represent genes that are transcribed in both samples.

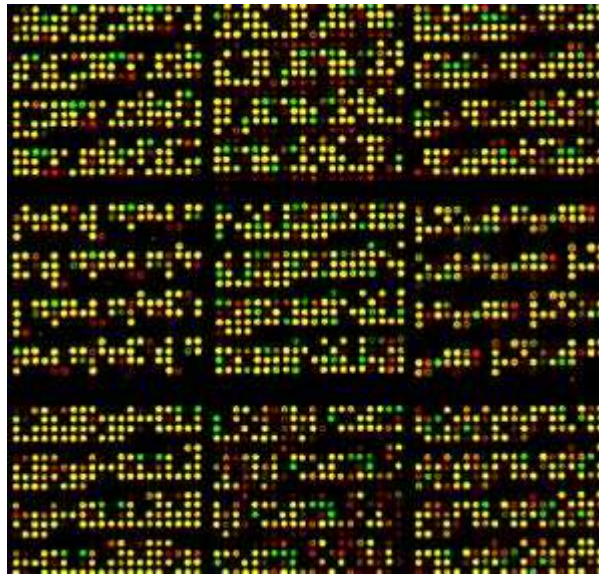


Figure 1.4 Spotted cDNA array

1.4. Pre-processing, Background Correction and Normalization

The arrays were scanned using the *Genetix Qscan* microarray reader a*QUIRE*. In order to obtain a single overall intensity value for each probe, the corresponding spots need to be identified. There are many methods available that are implemented in special software packages. We used the software *Genetix Qscan* for the so-called Spot Finding.

For this purpose the scanned image was loaded by the software together with a “Gene Array List” (GAL) file. The GAL-file is a text file with specific information for the layout of each block and the location, size, and name of each spot within a block on the microarray. Once the image and the GAL-file are loaded a grid appears on the image of the array and has to be moved so that every cycle of the GAL-file matches to the spot of the image (see Figure 1.5 d)). This can be done automatically but is very imprecise. As the accuracy of the Spot Finding has an impact on the correctness of the results it is better to do it manually. Each cycle in each block has to be adjusted in that way that no background pixels fall into the cycle, as this decreases the intensity of the spot. Figure 1.5 c) shows a perfectly adjusted spot.

After the Spot Finding is finished statistics are generated by the software. It contains a lot of information about the foreground and background intensities together with many other data of each spot. These statistics can be stored in a text file which is used for further steps of the analyses.

One important step in the analysis of microarray expression data is the quality check and the background correction of the array and normalization of the retrieved data. The quality check gives a hint if the array is good enough for further analyses or maybe should be repeated. The background correction is necessary before the normalization step because of fluctuation of the background within the array. It is generally performed by simply subtracting the observed background signal based on measurement made in the neighborhood of the DNA spot from the observed foreground signal in the DNA spot. There can be problems with negative values, just when the background value is greater than its foreground value (Mount 2004). For our

analyses we used the “minimum” method, which is available in MuMAT. This method sets all intensities that are zero or negative after subtraction of the background to half of the smallest positive intensity after subtraction of that array.

The aim of normalization is to compensate variation in the data points because of for example uneven variation of the two labels on one slide or between slides. Also variations in the spotting, sample preparation, labeling, or hybridization account for normalization before any analyzing step.

There are various normalization methods implemented in MuMAT. In our analyses for the within array normalization we used the “printploess” method and for the between array normalization the “quantile” method.

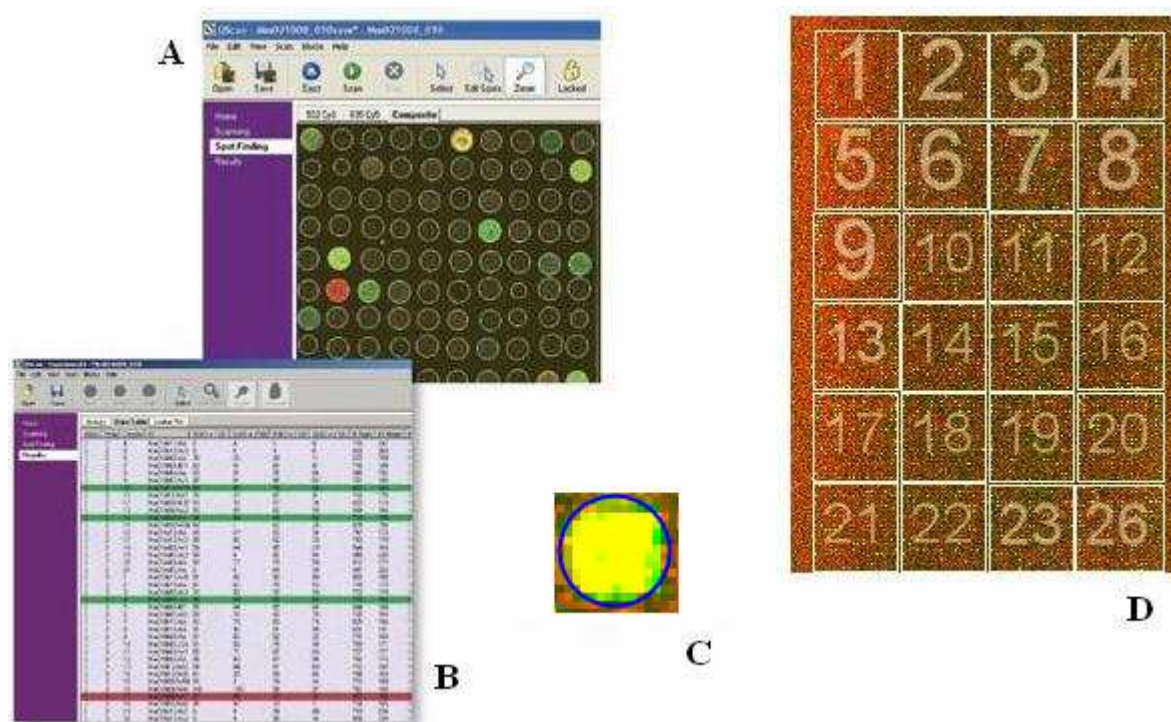


Figure 1.5 Genetix Qscan. **A** Spot Finding using the QScan Software. **B** QScan data table showing data colour coded by groups. Each data row is linked to the spot image for that data. **C** single spot. **D** First 26 blocks of an array.

[www.genetix.com]

2. Materials and Methods

Once the expression profiles are generated, the Spot Finding, the background correction, the normalization of the data and the BAGEL run is done, MuMAT provides several functions for the analyses of the retrieved data. As input for all tools the output file of the BAGEL run is necessary. For some methods an extra randomized BAGEL output file is needed. The files that are generated by MuMAT can be downloaded and renamed by the user. The way the tools are used is described in chapter 3.

2.1. *Drosophila melanogaster* lines

Dutch and Zimbabwean populations of *Drosophila melanogaster* were taken for the experiments. Eight lines per population were taken to analyze the expression variation between different lines.

2.2. Experimental design

To determine the difference of gene expression within the Dutch and Zimbabwean population as well as between the two populations, the experiment was designed as shown in Figure 2.2. As direct comparisons lead to more accurate results we tried to maximize them while keeping the total number of hybridizations at a practicable level and included unbroken chains of comparisons between all lines. This ensures that relative expression levels can be inferred of genes between samples that are not directly compared in competitive hybridizations (Meiklejohn and Townsend 2005). To reduce technical variation a dye-swap was performed for each experiment.

Figure 2.1 shows the hybridizations done in each of the populations indicated by arrows. Arrowheads symbolize the Cy3 labeled lines and the other side of the arrow the Cy5 labeled ones.

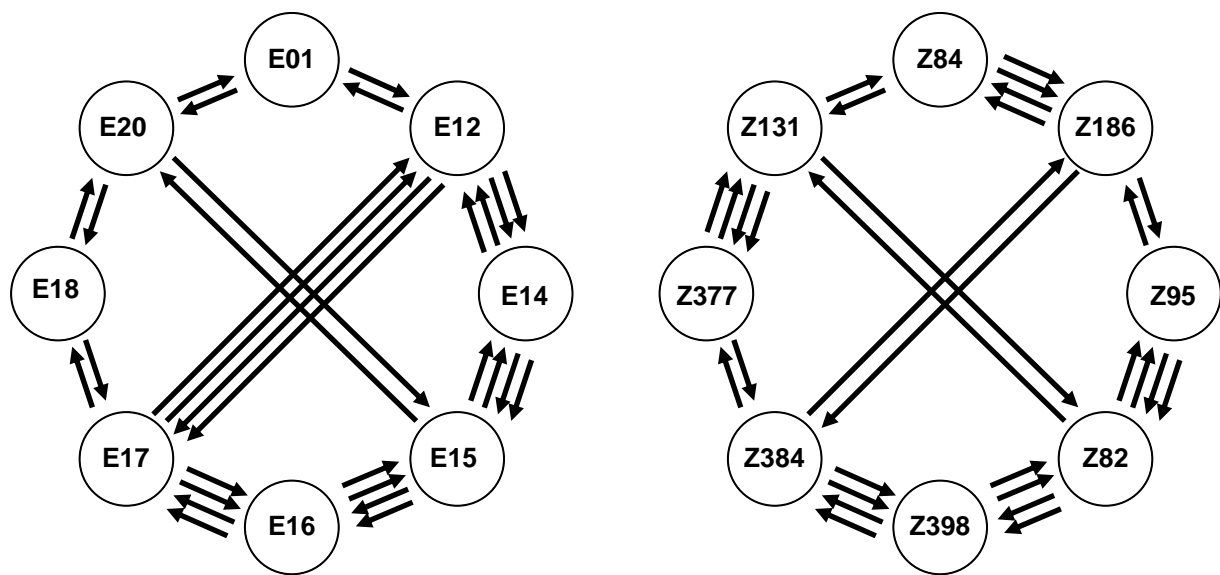


Figure 2.1 Left: Hybridizations between lines of the European population. Right: Hybridizations between lines of the African population

Figure 2.2 demonstrates the hybridization scheme of the total experiment, where grey arrows indicate the hybridizations within the populations, black arrows the hybridizations between populations. This allows analyses of the different expression profiles within each population as well as between the populations. For the inter-population comparison 20 hybridizations and for each within-population comparison 30 hybridizations were performed which results in a total number of 80 hybridizations.

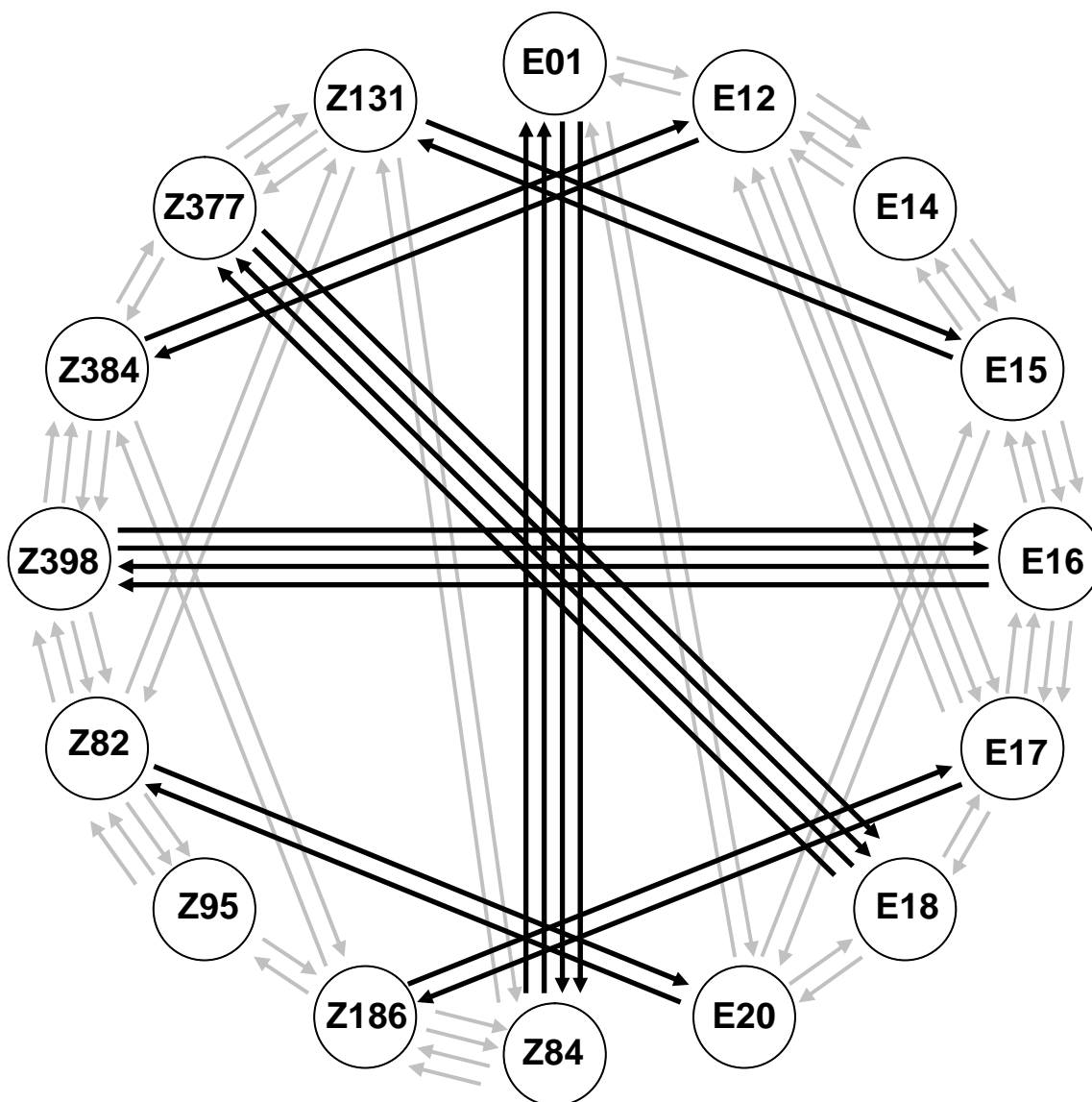


Figure 2.2 Hybridizations in the total experiment, where grey arrows indicate the hybridizations within the populations, black arrows the hybridizations between populations.

2.3. BAGEL

Bayesian Analysis of Gene Expression Levels (BAGEL) is a stand-alone software program that allows Bayesian analysis of gene expression levels on MacOs, Linux or Windows platforms. It accepts tab-delimited text files of ratio data as input (see Figure 3.2) and outputs a tab-delimited text file that displays in every line results for a single gene including columns id, name, the estimate of expression level for each expression node, where the levels are given relative to the sample with the lowest level, additions for 95% upper-bounds, and subtractions for 95% lower-bounds, coefficient of variation, posterior probabilities for whether that gene's expression level is greater in one sample than another, stationary acceptance rates for the Monte Carlo steps for that gene, and an "Acceptable?" column that gives information on whether those rates are acceptable, as well as Ln Density of Likelihood column (see Figure 3.6).

BAGEL uses a Markov Chain Monte Carlo based approach to determine the likelihood function of observed gene expression ratios. This method starts with a random number of parameters and then changes one or more of the parameters randomly. At each step the likelihood of the data given the model and the parameter values is calculated. If the new parameters give a better fit to the data, then the new values are accepted, otherwise the new values are accepted with a probability proportional to the ratio of the likelihood of the data with the new parameter values to the likelihood of the data with the old values (Meiklejohn and Townsend 2005). This is repeated as long as it converges from the initial parameter settings to a stationary distribution. Relative expression levels and statistical significance are inferred by sampling parameter values from the Markov chain (Townsend and Hartl 2002).

2.4. BAGEL randomizer

The BAGEL randomizer was already implemented in MuMAT, but two more options were added that enable the user to permute the BAGEL input file in three different ways: randomizing the columns, rows or the whole data matrix. Permuting the data is performed by sampling with replacement, except with the “randomize row” option, where sampling without replacement is done. In this case just the positions of the values within each gene are shuffled. The resulting randomized BAGEL input file has the same comparison structure as the original BAGEL input file.

2.5. BAGEL summarizer

2.5.1. Significant genes

This method of the BAGEL summarizer calculates the number of genes that are significantly expressed between samples and provides them in a table. For this calculation a randomized BAGEL (see section 2.2) output file is needed additionally to the real data BAGEL output file. Out of these files only the p-value columns are necessary. As the BAGEL output file contains p-values for whether the expression level is greater in one sample than in the other and vice versa, only one of the two p-value columns are taken for the calculation, that is if there is a column $P(S1>S2)$ and $P(S2>S1)$ only the column $P(S1>S2)$ is taken. Both p-values sum up to one, hence one value can be obtained by subtracting the other from one. In the next step for each pairwise comparison of two samples the numbers of genes are counted where the p-value is below the chosen p-value cutoff or above 1- p-value cutoff. These genes are considered as significant for the respective comparison. The numbers of significant genes for each comparison are written into a table. This is done for the real data file and the randomized data file. In the end for each pairwise comparison the numbers of significant genes of the randomized data is given above the diagonal of the data matrix and the numbers of significant genes of the real data below the diagonal. The names of the samples are shown in the header row of the table as well as in the left hand side of the table.

In addition to the table a summary is calculated. This comprises the average over all numbers of significant genes of the real data, the average over all numbers of significant genes of the randomized data and the False Discovery Rate (FDR). The FDR of a set of predictions is the

expected percent of false predictions in the set of predictions that is the randomized average divided by the real data average. The randomized data used for the calculation should have the same structure (number of expression nodes and comparisons between specific nodes) and proportion of missing data as the true data set, as these parameters will influence the FDR (Meiklejohn and Townsend 2005). The BAGEL randomizer preserves the structure of the real data and provides several opportunities for permuting the data.

All the computed data is available in a tab-delimited text file as well as in HTML. The name of the output file which is generated by this function is "SignGenes.txt".

2.5.2. FDR for each pairwise comparison

For the calculation of the FDR for each pairwise comparison of two samples the numbers of genes which are significantly differentially expressed for each pairwise comparison both for the randomized data and for the real data, as described in section 2.5.1, is needed. For this purpose each number of significant genes of the randomized data is divided by the number of significant genes of the real data. The results are then written into a table where the names of the samples are again shown in the top row and on the left-hand side of the table. Each cell of the table denotes the FDR for a pairwise comparison. The table is available in a tab-delimited text file as well as in HTML. The name of the output file is "PairwiseFdr.txt".

2.5.3. Number of up or down regulated genes

Also in this case only the p-values for the pairwise comparisons, where sample1 is greater sample2, were used. Then for each pairwise comparison of two expression nodes, the respective p-value column is passed, and if a p-value is below the preselected threshold, the second sample of the comparison is considered to be up-regulated, but if the p-value is above 1- threshold, then the first sample of the comparison is up-regulated, e.g. if $P(S1>S2) < p$ -value cutoff, then the gene is up-regulated in S2, but if $P(S1>S2) > 1-p$ -value cutoff, then the gene is up-regulated in S1. Then the number of genes is calculated where S1 is up-regulated, and where S2 is up-regulated. This is done for each pairwise comparison of the experimental nodes. All these numbers are then written into a table. For each pairwise comparison the number of genes with significantly higher expression in the sample given in the top row is shown above the diagonal. The number of genes with significantly lower expression in the sample given in the left-hand column is shown below the diagonal. Additionally a summary is shown that comprises the average number of genes in the up-regulated samples, the average number of genes in the down-regulated samples, and the ratio of up average value to down average value. All this data is available in a tab-delimited text file as well as in HTML. The name of the output file is "UpOrDownGenes.txt".

2.5.4. Up/down ratios for each pairwise comparison

The output of this method is a table, where the proportion of genes with differential expression that are up-regulated in the sample given in the top row is shown. Therefore for each pairwise comparison of two samples the number of genes up-regulated in the first sample of the comparison is divided by the number of genes which are down-regulated in the second sample. The table is available in a tab-delimited text file as well as in HTML. The name of the output file is "PairwiseUpDownRatio.txt".

2.5.5. List of significant differences per gene

In this case, a file that contains a table with three columns is generated. Each row displays the results for a single gene, including columns with, ID, name and a column for the number of comparisons, where the p-value is below the p-value cutoff or above 1- p-value cutoff. The name of the output file is "SignDiffPerGene.txt"

2.5.6. Fold change

For the calculation of the factor of gene expression difference, the columns with expression levels as well as the p-value columns of the BAGEL output file are needed. The p-value columns are searched for values that are below the p-value cutoff or above 1- p-value cutoff. If there's a hit, the expression levels of the respective samples are taken, and the factor of gene expression difference is calculated, that is, the bigger value is divided by the smaller one. All these fold change values are then added to a list, and the number of significant differences that fall in which fold change intervals are determined. The fold change intervals are defined in that way, that the maximum fold change value is taken out of the list and is rounded up to the next highest integer, and from 1 up to that maximum integer, values are generated at intervals of 0.5, e.g. if the maximum fold change is 2.78 then the following interval is generated: [1.0-1.5, 1.5-2.0, 2.0-2.5, 2.5-3.0] To calculate the numbers of significant differences that fall into the different intervals, counters for every interval are generated. The list of fold changes is passed, and for each value is checked into which interval it would fall. The respective counter is then increased by one. Furthermore the percentages of significant differences that fall into the different intervals are calculated. In the end a tab-delimited text file is created that contains a table, where the first column shows the fold change intervals, the second one the number of significant differences that fall into the particular intervals and the third column contains the percentage of significant differences that fall into the intervals. Below that table a summary is listed that shows the MSD_{50} and GEL_{50} for the p-value cutoff chosen by the user and the preselected 0.05 threshold.

The GEL_{50} and MSD_{50} can be used as a measurement of statistical power to detect expression differences. With this it is possible to compare the power of the experiment to earlier works.

The GEL_{50} is the factor of difference in gene expression that has a fifty percent chance of being identified as significant (Townsend 2004). Therefore, each fold change is assigned either one if the pairwise comparison of differential gene expression is identified as being significant or zero when identified as non-significant at the chosen cutoff level. A logistic regression on the factor of difference is computed from the data. This is done using the

algorithm LOGISTIC implemented in the software package WEKA (Frank 2005). It produces a maximum-likelihood curve describing the probability of calling an expression difference as significant given the best estimate of the fold change (Townsend 2004). The GEL_{50} is referred to as the fold change value at fifty percent probability.

For the determination of the median significant difference (MSD_{50}), the list of significant fold changes is sorted and the median fold change of that list is taken.

The name of the output file is "FoldChange_summary.txt".

Another text file is generated that contains a list of fold changes for each pairwise comparison of two samples. The output filename is "FoldChange_complete.txt".

2.6. Discrete expression states

This tool converts continuous gene-expression data to discrete states. On the one hand as states the bases A,T,C,G on the other hand alphabetical characters are possible depending on the parameters. Two methods are available, the Rank Neighbor and the Rank Reference method. With both methods the number of different expression states can be limited to four. In this case the states 'A','T','C','G' are used in this order, otherwise alphabetic characters are taken. Both provide the opportunity to use only genes with a CG number for the calculations. The Reference ranking but not the Neighbor one can be done in a descending or ascending order. The reasons are given below. In the end four files are created:

- One tab-delimited text file that contains in each row the name and id of the gene and the expression state for every sample. The last column gives the number of different states for the respective gene.
- One text file in fasta format that contains for every sample the discrete expression states for every gene in a sequence. Depending on the choice of number of states, the sequence is a nucleotide or character sequence, respectively.
- One text file in nexus format with Windows end-line characters that contains for every sample the discrete expression states for every gene in a sequence. Depending on the choice of number of states, the sequence is a nucleotide or character sequence, respectively.
- One text file in nexus format with Mac end-line characters that contains for every sample the discrete expression states for every gene in a sequence. Depending on the choice of number of states, the sequence is a nucleotide or character sequence, respectively.

2.6.1. Rank Neighbor

With limited number of states, the following procedure is performed: The expression levels of each sample are adapted consecutively for every gene. If we consider one gene all relative expression values of that gene are ordered ascending. The line with the lowest relative expression value -usually 1.0 - is taken as reference and the state 'A' is assigned to this line. The line with the next lowest expression value is taken as test line. If the p-value of the comparison reference vs. test is below threshold or above 1- threshold, the comparison is

considered to be significant, and the next state is assigned to the test line (e.g. 'T','C','G'). If there already exist four different states for that gene, all remaining lines are assigned state 'G'. If the comparison is not significant the same state as reference is assigned to this line. The test line is then the new reference. The next lowest expression value to the new reference is taken and the procedure is repeated as explained above. This is done until all expression values of that gene are replaced by discrete expression states. In the next steps the expression values of all genes are adapted in this manner. With unlimited number of states, the same procedure as above is performed except that as expression states alphabetical characters are taken and every time a new state is selected if the comparison reference vs. test is significant, without attending to the number of states.

With this method the order of comparison doesn't matter because just consecutive expression levels are compared as shown in Figure 2.3

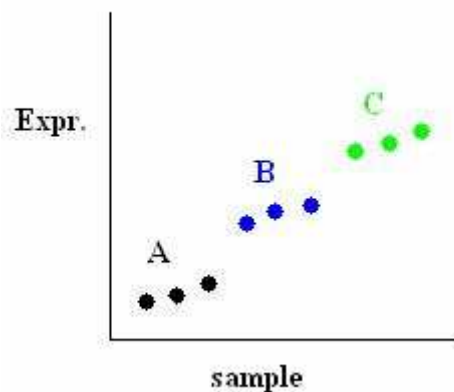


Figure 2.3 Rank Neighbor

2.6.2. Rank Reference

This method works like the method explained above, but in this case the reference line is not changed until a significant difference is found in the comparison of the p-values of the reference and the test line. If a significant difference is found, the test is then the new reference. With ascending order, it is started with the line with the lowest expression value and step by step the next lowest expression values are replaced by states, whereas with descending order, the first reference is the line with the highest expression value and in each step the states for the next highest expression values are determined. Thus the order of comparison has an impact on the assignment of the states. This is illustrated in Figure 2.4.

Each plot represents an expression value for the respective sample. The order of comparison is indicated by arrows. In Figure 2.4 a) state 'C' is assigned to two samples, whereas in Figure 2.4 b) three samples are assigned state 'C'. This can be explained by the fact that with descending order the example value highlighted in red is compared to the reference with state 'C' but with ascending order it is compared to the reference with state "B" and in each case the according state is assigned.

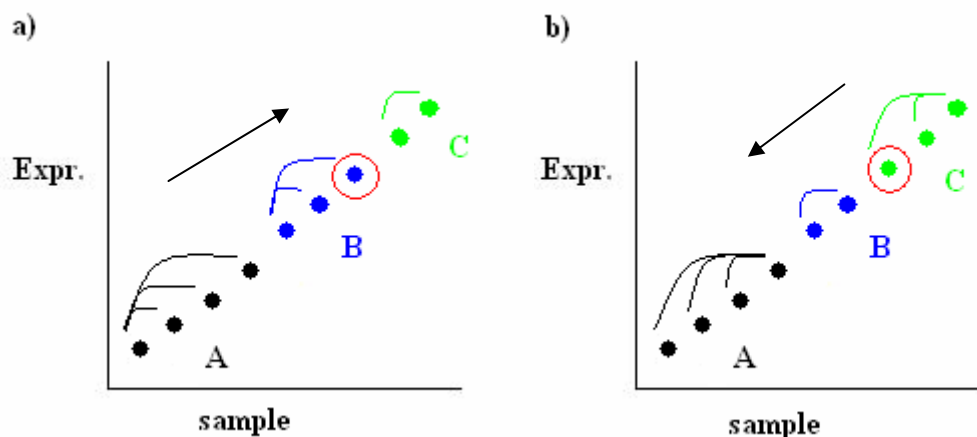


Figure 2.4 Rank Reference a) ascending order b) descending order

2.7. PAUP*

We used version 4.0b of the PAUP* (Phylogenetic Analysis Using Parsimony) software to group different lines, used for our experiments, based on their gene expression similarities. PAUP* is a widely used tool for the inference of evolutionary trees. It is available for Macintosh, Windows, UNIX/VMS, or DOS. The data file format used by PAUP* is a nexus-format and is one of the data file formats provided by the “expression states” function in MuMAT (see section 3.4).

2.7.1. Parsimony Tree

PAUP* 4.0 has the possibility to analyze data using several different optimality criteria: parsimony, likelihood, and distance. We used the maximum parsimony criterion to search for optimal trees, which is the default setting. For searching for optimal trees PAUP* provides two basic classes of methods: exact and heuristic. Exact methods guarantee to find the optimal tree(s) but may require prohibitive amounts of computer time for medium to large-sized data sets. Heuristic methods do not guarantee optimality but generally require far less computer time. For our analysis we used the heuristic search method.

Additionally we performed a bootstrap analysis. The method involves sampling the original data set with replacement to construct a series of bootstrap replicates of the same size as the original data set. Each of these is then analyzed using either a heuristic search or branch-and-bound. Finally, a majority-rule consensus tree is constructed for all of the bootstrap trees. If a group appears in X percent of the bootstrap trees, the confidence level associated with that group is taken as X percent. This method gives the ability to assign statistical confidence to hypotheses of relationship.

We defined 200 bootstrap replications (resamplings) and a tree search is performed for each resampling using heuristic search. The output of the bootstrap procedure consists of a table showing all partitions (or groups) that were found in the bootstrap replications and their


frequencies (see Figure 2.5), and a bootstrap majority-rule consensus tree. The numbers on the branches of the consensus tree indicate the percentage of the bootstrap replications that support the group descending from that branch. The frequencies indicate the number of bootstrap replicates in which the particular partition was found.

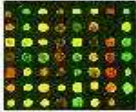
Partitions found in one or more trees and frequency of occurrence:

12345678	Freq

...*..**	91.01
.....**	78.66
..*****	77.66
...*****	63.40
...*...***	42.83
..**...***	25.80
..**..**..	8.80
.....***	8.79
.....**..	4.34
..**.....	4.07
.*...*...*	3.09
..*...*..	2.39
..*...**..	0.78
..**.*...*	0.40
..*...*...*	0.33
..**...*...*	0.30
.*...**..	0.30
..*...*****	0.17
..*...***	0.12

Figure 2.5 Groups that were found in the bootstrap replications and their frequencies





- home
- normalize
- bagel randomizer
- average over genes
- download
- bagel extractor
- bagel summarizer
- expression states
- help

home

short description:

1. **normalize:**
input: genetix qscan statistic file
output:
 - ◊ quality check
 - ◊ gpr format
 - ◊ normalized values
 - ◊ several plots (pdf format)
 - ◊ xls file containing further information useful for creating a bagel input file
2. **bagel randomizer:**
input: bagel input file
output: randomized bagel input file
3. **average over genes:**
input: file containing gene names in the first column and any kind of numeric values in the other columns
output: file containing same gene list but genes are unique now values are averaged for former multiple occurrence of gene name
4. **bagel extractor:**
input: bagel output file
output: file that contains selected columns of the bagel output file
5. **bagel summarizer:**
input: bagel output file and randomized bagel output file
output:
 - ◊ table of significant genes
 - ◊ table of FDRs
 - ◊ table of number of up and down regulated genes
 - ◊ table of up/down ratios
 - ◊ list of significant differences per gene
 - ◊ table of fold changes, a list of all fold changes, a histogram and a table that shows the GEL50 and MSD50
6. **expression states:**
input: bagel output file
output: file in four different formats that contains discrete gene-expression states

for details see [help](#)

Figure 3.1 MuMAT

3. MuMAT

Figure 3.1 shows the home page of MuMAT, accessible within the LMU Biozentrum. As already mentioned in the previous sections MuMAT is a selection of several functions for the analyses of microarray expression data. The toolbox on the left-hand side shows all tools that are available. Some of them had already existed on the home page, some were added in the course of this thesis. In the frame on the right hand side a short description of all methods available in MuMAT is given. In this chapter I will introduce the new functions and give a brief tutorial of how to use them.

3.1. BAGEL randomizer

As already mentioned in the previous chapter, for several statistical computations like the number of false positives, a randomization of the normalized data is necessary. There are three different options for randomly permuting the data, implemented in the BAGEL randomizer.

First the input file for BAGEL has to be uploaded which should be randomized (see Figure 3.3). BAGEL accepts tab-delimited text files with three header rows. The second and third rows must contain unique names for each experimental expression node and reference expression node, followed by any number of data rows for each gene of interest (see Figure 3.2).

[Your Notes]	[Your Notes]	[Label1]	[Label2]	[Label3]...
[Your Notes]	[Channel1]	Exp1	Exp2	Exp3 ...
[Your Notes]	[Channel2]	Ref1	Ref2	Ref3 ...
ORF1	CommonName1	Ratio1	Ratio2	Ratio3 ...
...

Figure 3.2 Input file for BAGEL

home

normalize

bagel randomizer

average over genes

download

bagel extractor

bagel summarizer

expression states

help

randomize

upload your bagel input file

note! the output file will have WINDOWS end line characters
MAC or LINUX user have to convert the end line characters to use BAGEL!

Figure 3.3 BAGEL randomizer - upload

randomize

Your uploaded files

european.txt

number of randomizations

randomize columns

randomize rows

randomize whole matrix

! the randomization may take a few minutes

Figure 3.4 BAGEL randomizer - settings

In the next step the user can define the number of randomizations. As default, the number of genes of the data matrix of the input file is set. This makes sense as the permuted data matrix should have the same structure as the true data set. Afterwards the type of randomization has to be chosen from three different options, randomizing the columns, rows or the whole data matrix (see Figure 3.4). In the end the resulting file can be downloaded and used as input for BAGEL (see Figure 3.5). Clicking the “get result” button will bring up a standard save file dialog window.

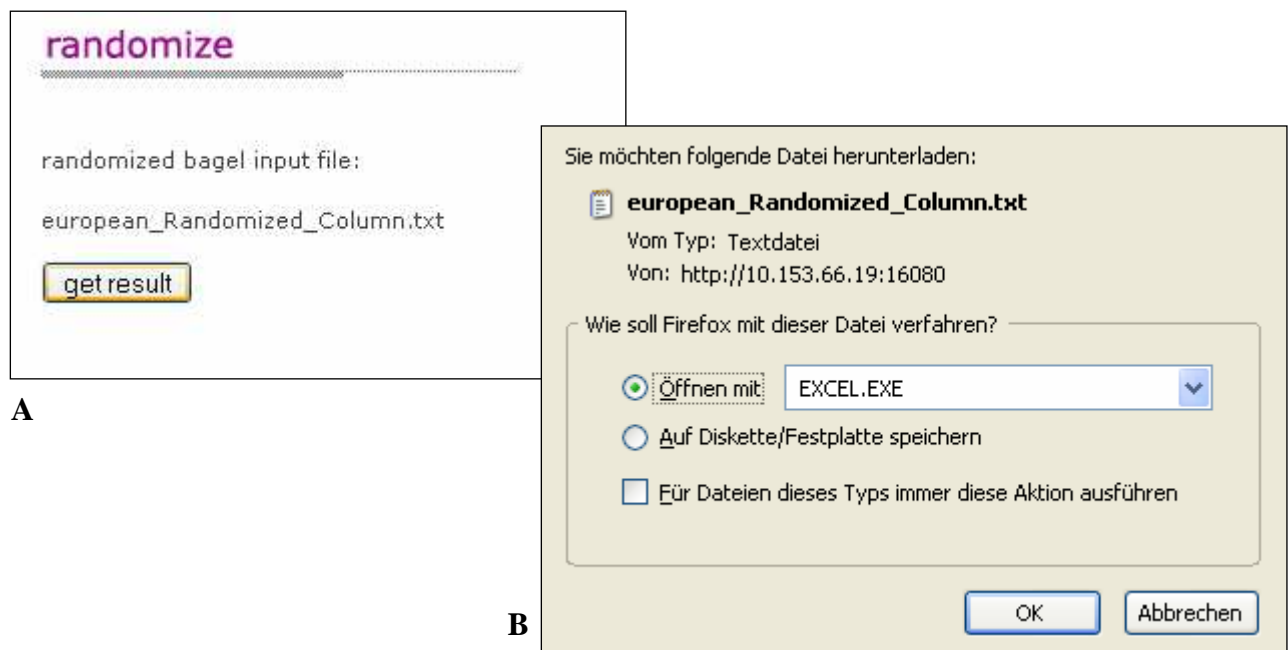


Figure 3.5 BAGEL randomizer – **A:** download the result **B:** download dialog

3.2. BAGEL extractor

The BAGEL extractor extracts several columns, which can be selected by the user, out of the BAGEL output file and writes them into a tab-delimited text file. The name of the output file is the same as the original filename, but has the characters “_extract.txt” appended.

As already mentioned, the first ten lines of the BAGEL output file contain the parameter settings. The following lines show the columns id, name, expression levels for each expression node, additions for 95% upper- bounds, and subtractions for 95% lower-bounds, coefficient of variation, p-values for whether expression level is greater in one sample than another, Mu and Variance/CV step acceptance rate, “Acceptable?” column, as well as Ln Density of Likelihood column (see Figure 3.6). The levels of gene expression are given relative to the one with the lowest expression level (lowest value is always 1).

gene name						
Unique ID	Common Name	CS	CSf	Sim	Simf	...
PGRP-SC1b	GH07464	1	1.17	2.37	2.85	...
BcDNA:LD09936	LD09936	11.35	1.78	22.10	1	...
CG12200	LD30246	1.28	7.13	1.43	1	...
qtc	SD06355	2.59	1	2.65	1.11	...
...						
(-)97.5%[CS]	(-)97.5%[CSf]	(-)97.5%[Sim]	(-)97.5%[Simf]			...
0.32146	0.34871	0.38685	0.39713			
1.055	0.94196	1.52557	0.79782			...
0.20477	0.48235	0.28134	0.18666			...
0.18859	0.17951	0.18848	0.17595			...
...						
P(CS>CSf)	P(CS>Sim)	P(CS>Simf)	P(CSf>CS)	P(CSf>Sim)		...
0.2543	0.0001	0	0.7457	0.0004		...
1	0	1	0	0		...
0	0.1591	0.9642	1	1		...
1	0.3336	1	0	0		...

Figure 3.6 BAGEL output file

The execution of the BAGEL extractor is pretty easy. First the BAGEL output file, from which the data should be written into another file, has to be uploaded. The file which has been uploaded is displayed and by clicking the “extract” button all columns which can be extracted are shown. (see Figure 3.9).

There are several groups of columns that can be selected for extraction. The *names* include the first two columns of the BAGEL output file, the “UNIQUE ID” and the “COMMON NAME” column, the *expression values* denote the estimates for each expression node and the *rest* the four columns at the end of the BAGEL output file. Other groups are the *confidence intervals*, *coefficient of variation* and *p-values*. There are two possible alternatives for the p-values, on the one hand all p-values on the other hand *p-values only one direction*. The latter indicate only one p-value column for each comparison of two expression nodes. For each comparison there are two p-values that sum up to one, thus one p-value can be calculated by subtracting the other from one.

Next to each group the number of columns for the respective group is given. The sum of all selected columns is given at the bottom. This helps users that want to use Excel to open the created file to decide correctly on the number of columns, since Excel can only handle at most 256 columns.

If the checkbox “input for clustering” is selected, an input file for the Clustering algorithm by Eisen *et al.* (1998) is generated.

In the end the output of the BAGEL extractor can be downloaded. (see Figure 3.10).

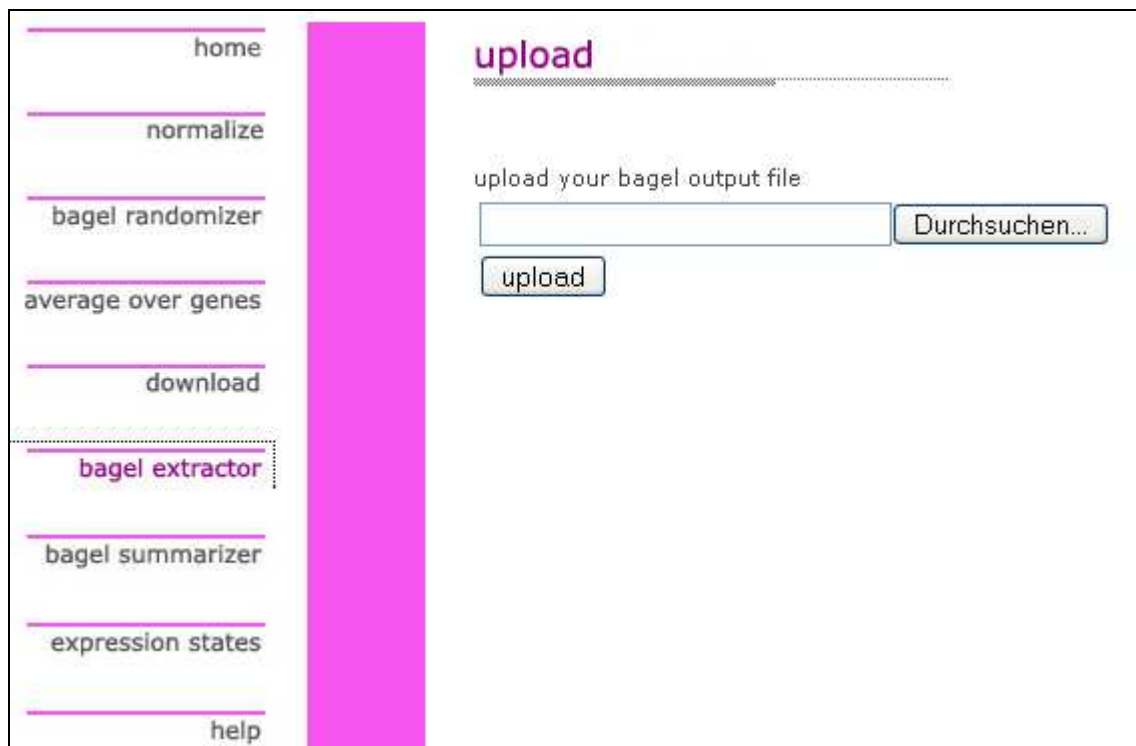


Figure 3.7 BAGEL extractor - upload

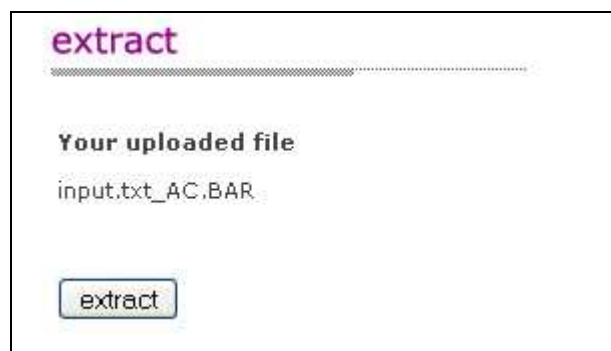


Figure 3.8 BAGEL extractor – extract

extract

Mark all you want to be written into one file!
Note that if you want to use Excel to open the file the number of columns should not exceed 256!

names	<input checked="" type="checkbox"/>	2
expression levels	<input checked="" type="checkbox"/>	2
confidence intervals	<input type="checkbox"/>	4
coefficient of variation	<input type="checkbox"/>	6
p-values all	<input type="checkbox"/>	2
p-values one direction only	<input type="checkbox"/>	1
rest	<input type="checkbox"/>	4
sum of columns		<input type="text" value="4"/>

Mark the checkbox if you want to create an input file for clustering!

input for clustering

Figure 3.9 BAGEL extractor - settings

extract

◆ **file extraction:**
input_extract.txt

◆ **file for clustering:**
input_cluster.txt

Figure 3.10 BAGEL extractor - download

3.3. BAGEL summarizer

The methods of the BAGEL summarizer are explained in detail in chapter 2. Here I will just focus on the easy handling of this function. As input for the BAGEL summarizer serve the BAGEL output file of the real data together with the BAGEL output file of the randomized data (see Figure 3.11). The format of these files is shown in Figure 3.6. As shown in Figure 3.12 a threshold in the format $x.xxx$ has to be typed in and if desired the checkbox “only CG genes” has to be checked. If this checkbox is marked, only the genes of the input files that have a CG-number are taken for calculations. This makes sense if DGRC-1 arrays have been used for experiments, otherwise there would exist no genes with CG numbers in the BAGEL output file. Hence, if other array types have been taken, this checkbox should not be marked. By clicking the “summarize” button the BAGEL summarizer outputs several information (see Figure 3.13):

- table of number of genes with a significant comparison in a tab-delimited text file and as HTML
- table of false discovery rates for each significant pairwise comparison of two samples, in a tab-delimited text file and as HTML
- table of numbers of genes for each pairwise comparison with significantly higher expression in one of the two lines, in a tab-delimited text file and as HTML
- table that shows for each pairwise comparison ratio of numbers of genes with significantly higher expression in one of the two lines, in a tab-delimited text file and as HTML
- list of significant differences per gene
- two files with information about the fold changes, a histogram and a table that shows the GEL_{50} and MSD_{50} for two p-value cutoffs

home

normalize

bagel randomizer

average over genes

download

bagel extractor

bagel summarizer

expression states

help

upload

upload your bagel output file

upload your randomized bagel output file

Figure 3.11 BAGEL summarizer - upload

summarize

Your uploaded files

all_negdistdna_95.txt_AC.BAR

all_negdistdna_95_CG_ran.txt_AC.BAR

p-cutoff value

note! Please type in a number in the following format: x.xxx

only CG genes

 the summarization may take a few minutes

Figure 3.12 BAGEL summarizer - settings

summarize

- ◆ **significant genes**
 -
- ◆ **FDR for each pairwise comparison**
 -
- ◆ **up or down regulated genes**
 -
- ◆ **up/down ratio for each pairwise comparison**
 -
- ◆ **list of significant differences per gene**
 -
- ◆ **fold change**
 - ◇ summary list
 -
 - ◇ complete list
 -
 - ◇ summary statistics

P-value	0.001	0.05
GEL 50	2.12973	1.4192
MSD 50	1.35064	1.22468

Figure 3.13 BAGEL summarizer - output

Figure 3.13 shows the output created by the BAGEL summarizer. The way this output is generated is described in chapter 2. By clicking the “get file” button the respective files can be downloaded, and by clicking the “show table” button the data of the output file is shown as HTML. Figure 3.14 shows the table and summary for the “number of significant genes”. The names of the expression nodes are given in the header row of the data table as well as in the left hand side. Additionally to the data table a summary is calculated. Such a table and

summary is generated additionally for the “false discovery of pairwise comparison”, “up- or down regulated genes” and the “up/down ratio of each pairwise comparison”.

The “list of significant differences per gene” is a file that contains a table with three columns. Each row displays the results for a single gene, including columns with, ID, name and a column for the number of comparisons, where the p-value is below the chosen p-value cutoff or above 1- p-value cutoff.

	MEL01	MEL12	MEL14	MEL15	MEL16	MEL17	MEL18	MEL20
MEL01		21	6	11	15	9	15	14
MEL12	172		15	13	37	19	20	14
MEL14	120	309		8	23	10	22	6
MEL15	135	249	269		29	11	17	13
MEL16	115	185	202	197		18	44	16
MEL17	146	396	257	326	436		19	24
MEL18	94	119	118	134	99	167		12
MEL20	152	182	148	184	149	178	165	

p-value cutoff	0.0010
data average	192.96428571428572
random average	17.178571428571427
FDR	0.08902461595409956

Figure 3.14 BAGEL summarizer – table of significant genes

As described in the previous chapter, the “fold change” provides information about the factor of expression difference of two samples whose comparison of the p-values is significant. Here, as already mentioned, two files one histogram and one table is generated.

One of the files, named "FoldChange_summary.txt", contains a table, where the first column shows the fold change intervals, the second one the number of significant comparisons that fall into the particular intervals and the third column contains the percentage of significant differences that fall into the intervals (see Figure 3.15). Below that table a summary is listed that shows the MSD₅₀ and GEL₅₀ for the p-value cutoff chosen by the user and the preselected 0.05 threshold.

The MSD₅₀ and GEL₅₀ are also given in a table on the output page as shown in Figure 3.13.

The second file contains the complete list of fold changes in one column. With this file it is possible for the user to create his own histograms and to make further analyses. The output filename is "FoldChange_complete.txt".

All the output files generated by the BAGEL summarizer can be downloaded. The preselected file name can be changed by the user.

fold change	number of significant genes	%
1.0 - 1.5	86	72.27%
1.5 - 2.0	24	20.17%
2.0 - 2.5	4	3.36%
2.5 - 3.0	1	0.84%
3.0 - 3.5	0	0.00%
3.5 - 4.0	1	0.84%
4.0 - 4.5	1	0.84%
4.5 - 5.0	0	0.00%
5.0 - 5.5	2	1.68%
5.5 - 6.0	0	0.00%
total	119	
P-value cutoff:	0.001	
Msd 50:	1.35064	
Gel 50:	2.12973	
P-value cutoff:	0.05	
Msd 50:	1.22468	
Gel 50:	1.4192	
filename of data:	inter_qual.txt_AC.BAR	

Figure 3.15 BAGEL summarizer – fold change summary

To demonstrate the factor of gene expression difference graphically, a histogram is constructed. This can be viewed by clicking the “show histogram” button. The x-axis shows the fold change intervals and the y-axis the percentage of significant differences that fall into the respective intervals. A cutoff for the x-axis is set to limit the complexity of the histogram. If there are three percentage values in a row that are less than one percent, this interval is set as a cutoff, and all succeeding intervals are summarized in that interval, that means, all succeeding percentage values are summed up and displayed in the cutoff interval. Figure 3.16 shows such a histogram, where the cutoff for the x-axis is 5.0, thus all percentage values of significant genes that fall into intervals that are bigger than 5.0 are pooled in that interval.

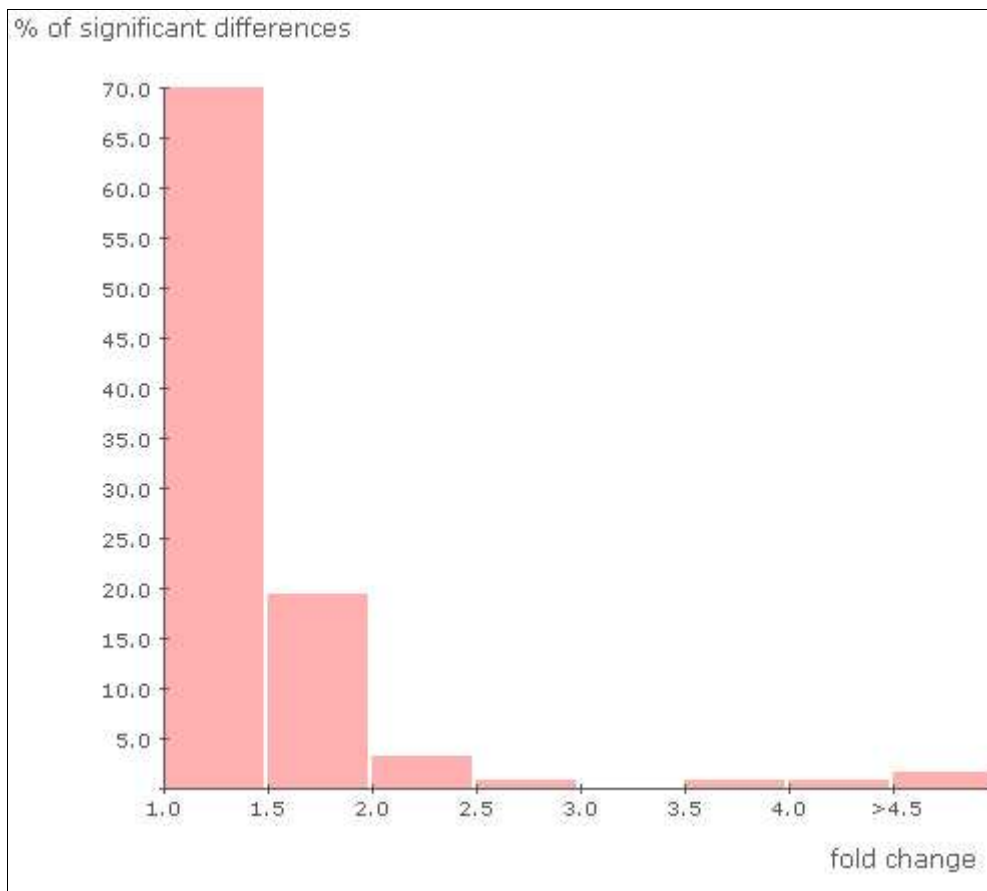


Figure 3.16 BAGEL summarizer - histogram

3.4. Discrete expression states

To use the function discrete expression states, the BAGEL output file has to be uploaded as shown in Figure 3.17.

There are two methods, as already mentioned in chapter 2, the *Rank Neighbor* and *Rank Reference* method. For each of them several options can be chosen.

With both methods it is possible to limit the number of different gene expression states to four. In this case the states 'A','T','C','G' are used in this order. If this option is not selected alphabetical characters are used as states. The alphabetic characters are those UNICODE characters which are defined as letters by the UNICODE standard, e.g., the ASCII characters 'A' 'B' 'C' 'D' 'E' 'F' 'G' 'H' 'I' 'J' 'K' 'L' 'M' 'N' 'O' 'P' 'Q' 'R' 'S' 'T' 'U' 'V' 'W' 'X' 'Y' 'Z'.

The *Rank Reference* method also provides the option to decide about the order of comparison. In either case the p-value cutoff has to be entered in the text field and the user has to decide whether only genes with CG numbers should be incorporated or not. In this case the respective checkbox has to be selected (see Figure 3.18).

home

normalize

bagel randomizer

average over genes

download

bagel extractor

bagel summarizer

expression states

help

upload

upload your bagel output file

Figure 3.17 discrete expression states - upload

discrete

Your uploaded file
all_negdistdna_95.txt_AC.BAR

p-cutoff value

note! Please type in a number in the following format: x.xxx

Rank Neighbor

limit to four expression states

Rank Reference

limit to four expression states

descending ascending

only CG genes

Figure 3.18 discrete expression states - settings

The output filename is the same as the original input filename but has the characters “_RankNeighbor.txt” or “_RankReference.txt” appended, depending on the type of method selected. Additionally three different formats of the file are generated: Fasta-format and Nexus-format in Windows and Mac version. These files have additionally the characters “Fasta.txt”, “NexusWin.txt” or “NexusMac.txt” appended.

Figure 3.19 shows the four files that have been created by MuMAT:

- One tab-delimited text file that contains in each row the name and id of the gene and the expression state for every sample. The last column gives the number of different states for the respective gene.
- On text file in fasta and nexus format with Windows and Mac end-line characters, that contains for every sample the discrete expression states for every gene in a sequence. Depending on the choice of number of states, the sequence is a nucleotide or character sequence, respectively.

A sequence in fasta format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. An example sequence in FASTA format is:

```
>name of the sequence
ttctcccagaagctgactctatgngacccccgagagagactgagcagaacctggagccag
ccccgcaccctgcacttccaatcaggggccccgggagcactccccgtggcgccgcgc
ccgccctccgcgcagccatg
```

The nexus format is characterized by the symbol “#” followed by the word “nexus”. Subsequent to “begin data;” the description of the data sequence is listed, which gives the number of taxa, the length of each sequence together with the sequence type and the symbols for gaps and missing data. The data sequence is distinguished from the description by the word “matrix”. The symbol “;” and “endblock;” indicate the end of the file. An example sequence in nexus format is:

```
#NEXUS
BEGIN DATA;
  DIMENSIONS  NTAX=8  NCHAR=2;
  FORMAT DATATYPE=STANDARD  SYMBOLS="ABCDEFGH"  MISSING=?  GAP=-  ;
MATRIX

MEL01 CE
MEL12 DA
MEL14 EA
MEL15 AC
MEL16 BD
MEL17 CB
MEL18 FB
MEL20 CF
;
END;
```

Saving both formats makes it easy to use the results in programs like PHYLIP and PAUP*.

discrete

- ◆ **output file:**
all_negdistdna_95_RankNeighbor.txt
[download](#)
- ◆ **in fasta format:**
all_negdistdna_95_RankNeighborFasta.txt
[download](#)
- ◆ **in nexus format (with Windows end-line characters):**
all_negdistdna_95_RankNeighborNexusWin.txt
[download](#)
- ◆ **in nexus format (with Mac end-line characters):**
all_negdistdna_95_RankNeighborNexusMac.txt
[download](#)

Figure 3.19 discrete expression states - output

4. Results

4.1. Comparison of individual lines

To generate statistics about the different gene expression of all lines, we did two separate BAGEL runs in order to obtain BAGEL results for each pairwise comparison for the real data as well as for the randomly permuted data. Only "meaningful" red/green ratios based on a negative distribution for signal above background for each slide were used. As negative controls those spots were used that contain exogenic DNA (e.g. DNA from yeast). For each array the distribution of the signals above background for these negative controls was determined for both channels separately. Expressed spots were defined as signals which had a signal above background which was higher than the 95 percentile of the negative distribution for both channels.

To get a randomized data set the BAGEL input file containing the normalized ratios for each array was randomized using the BAGEL randomizer in MuMAT (see chapter 3.1). The number of genes in the randomized file was the same as in the original data file, otherwise this would influence the estimates. Randomization was performed by sampling with replacement within each hybridization.

Using the real data and the randomized data we calculated the FDR, that is the proportion of genes with statistically significantly different expression that are expected to be false positives (Meiklejohn and Townsend 2005). Table 4.1 shows the FDR at four different p-value cutoffs. As a p-value of 0.05 would lead to a very high number of false positives we decided to use a p-value of 0.001 as significance threshold which led to a FDR of about 6.9%.

P <	0.05	0.01	0.001	0.0001
FDR	49,02 %	23,34 %	6,87 %	2,65%

Table 4.1 FDR at four significance thresholds

In Table 4.2 the number of significantly differentially expressed genes for each pairwise comparison of two lines are given. Above the diagonal are the estimates for the randomized data and the numbers below the diagonal indicate the estimates for the real data set. The average number of genes that are significantly differentially expressed in the real data set is 137.98 and in the randomized data set 9.49. The variation in gene expression difference between the two populations is not very different. The average number of significant pairwise differences in the European population is 126.46 and in the African population 125.86.

Table 4.3 shows the FDR for each comparison of two lines. European line E15 compared to the African line A131 shows a relative high FDR of about 45%. Additionally for each pairwise comparison we calculated the number of genes that are significantly higher and lower expressed in one of the lines respectively. Table 4.4 shows above the diagonal the number of genes with significantly higher expression in the strain given in the top row. Below the diagonal the number of genes with significantly lower expression in the strain given in the left-hand column is shown. For each pairwise comparison the proportion of genes with differential expression that are up-regulated in the line in the top row is shown in Table 4.5.

	E01	E12	E14	E15	E16	E17	E18	E20	A84	A131	A398	A82	A186	A95	A384	A377
E01		9	2	7	6	11	7	6	9	8	5	10	17	13	14	12
E12	168		5	10	7	7	4	8	12	13	10	18	7	14	16	10
E14	74	151		8	7	3	4	2	10	9	5	3	6	9	3	8
E15	93	145	137		8	6	7	9	6	19	11	9	11	14	7	8
E16	99	111	91	76		4	3	7	9	7	15	9	5	13	10	5
E17	80	255	114	151	221		5	4	9	6	9	12	6	15	9	6
E18	91	99	92	96	98	94		5	7	4	4	9	6	12	5	8
E20	139	156	106	174	145	117	168		19	9	15	9	12	25	8	10
A84	180	132	108	79	97	110	79	154		10	9	16	15	23	14	10
A131	109	121	95	42	97	98	129	150	80		16	20	8	19	12	6
A398	180	222	161	145	274	157	110	245	66	93		6	7	12	4	4
A82	131	164	109	92	141	148	104	280	72	133	54		11	25	9	11
A186	118	147	93	83	110	52	105	165	167	105	109	89		23	6	6
A95	216	220	153	112	168	299	165	322	127	188	200	180	192		13	13
A384	128	228	123	135	196	160	148	187	112	105	164	148	157	240		7
A377	126	180	131	105	138	120	229	188	78	116	84	97	88	178	102	

Table 4.2 Numbers of differentially expressed genes for each pairwise comparison at a significance threshold of 0.001. Numbers below the diagonal are estimates from the original data, numbers above the diagonal are from the randomized data.

	E01	E12	E14	E15	E16	E17	E18	E20	A84	A131	A398	A82	A186	A95	A384	A377
E01																
E12	0.05															
E14	0.03	0.03														
E15	0.08	0.07	0.06													
E16	0.06	0.06	0.08	0.11												
E17	0.14	0.03	0.03	0.04	0.02											
E18	0.08	0.04	0.04	0.07	0.03	0.05										
E20	0.04	0.05	0.02	0.05	0.05	0.03	0.03									
A84	0.05	0.09	0.09	0.08	0.09	0.08	0.09	0.12								
A131	0.07	0.11	0.09	0.45	0.07	0.06	0.03	0.06	0.13							
A398	0.03	0.05	0.03	0.08	0.05	0.06	0.04	0.06	0.14	0.17						
A82	0.08	0.11	0.03	0.1	0.06	0.08	0.09	0.03	0.22	0.15	0.11					
A186	0.14	0.05	0.06	0.13	0.05	0.12	0.06	0.07	0.09	0.08	0.06	0.12				
A95	0.06	0.06	0.06	0.13	0.08	0.05	0.07	0.08	0.18	0.1	0.06	0.14	0.12			
A384	0.11	0.07	0.02	0.05	0.05	0.06	0.03	0.04	0.13	0.11	0.02	0.06	0.04	0.05		
A377	0.1	0.06	0.06	0.08	0.04	0.05	0.03	0.05	0.13	0.05	0.05	0.11	0.07	0.07	0.07	

Table 4.3 FDR for each pairwise comparison ($P < 0.001$)

	E01	E12	E14	E15	E16	E17	E18	E20	A84	A131	A398	A82	A186	A95	A384	A377
E01		122	37	47	60	52	48	67	122	65	111	78	79	130	63	79
E12	46		44	47	39	100	40	61	59	43	113	69	80	117	85	77
E14	37	107		67	61	68	51	51	71	56	94	53	67	94	58	74
E15	46	98	70		44	71	52	76	48	25	91	36	60	63	64	67
E16	39	72	30	32		108	51	65	60	57	172	67	65	104	100	78
E17	28	155	46	80	113		39	50	67	59	91	80	38	190	74	66
E18	43	59	41	44	47	55		68	48	69	60	53	67	93	71	127
E20	72	95	55	98	80	67	100		101	90	149	166	106	215	101	114
A84	58	73	37	31	37	43	31	53		44	36	24	106	59	42	41
A131	44	78	39	17	40	39	60	60	36		44	62	63	114	44	62
A398	69	109	67	54	102	66	50	96	30	49		34	58	121	74	43
A82	53	95	56	56	74	68	51	114	48	71	20		61	96	67	55
A186	39	67	26	23	45	14	38	59	61	42	51	28		103	65	45
A95	86	103	59	49	64	109	72	107	68	74	79	84	89		80	71
A384	65	143	65	71	96	86	77	86	70	61	90	81	92	160		62
A377	47	103	57	38	60	54	102	74	37	54	41	42	43	107	40	

Table 4.4 Number of genes with significantly higher expression in one of the two lines ($P < 0.001$). Above the diagonal: numbers of genes with significantly higher expression in the strain given in the top row. Below the diagonal: number of genes with significantly lower expression in the strain given in the left-hand column

We calculated the number of significant pairwise comparisons for each gene at the 0.001 significance level. Figure 4.1 shows that most genes have no or just one or two significant pairwise differences out of the 120 possible ones. Numerical, 62.49% of all genes have no significant comparison, whereas 9.02% have one and 6.30% have at least 16 significant comparisons. The gene with CG number CG1180 has 73 pairwise differences. That is the highest number detected.

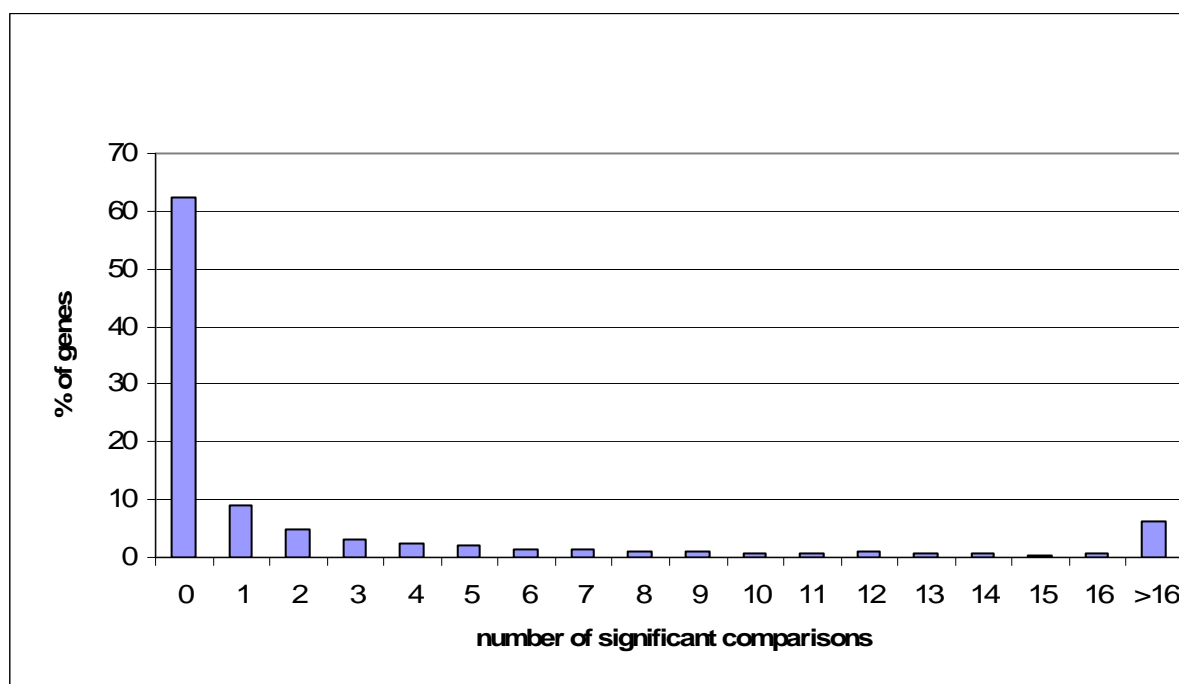


Figure 4.1 Number of significant pairwise comparisons for all genes

Figure 4.2 shows a histogram that plots the frequency of a significant call against the factor of gene expression difference. Most significant differences in expression between two lines that is 46.38% have a fold-change between 1.5 and 2.0, whereas 23.03% have a fold-change between 1.0 and 1.5 and 15.36% between 2.0 and 2.5. The smallest detected fold-change was 1.11 and the biggest one 36.55. This gives an idea about the power to detect true differences in gene expression. But to compare the power of our experimental design to other works we calculated the GEL_{50} . We measured a GEL_{50} of 1.53 at the 5% significance level which means that there is a 50% empirical probability obtaining a significant expression difference of 1.53-fold at a 5% significance level.

Study	Organism	GEL_{50}	Experimental design (Hybs/Nodes)
Meiklejohn et al. (2003)	Fruit flies	1.64	23/8
Ranz et al. (2004)	Fruit flies	1.15	15/3
Townsend et al. (2003)	Yeast	2.00	10/4
Brem et al. 2002	Yeast	3.82	23/13

Table 4.6 GEL_{50} of four different citations ($P < 0.05$)

Compared to the other studies listed in Table 4.6, the study by Ranz *et al.* (2004) had the finest resolution, detecting significant changes as small as 1.15-fold. This can be explained with the fact that they had five times as many hybridizations as nodes (Clark 2007). Although this experimental design and the one we used in our studies (hybridizations/nodes = 80/16) have same experimental design power the result for our experiment indicate a lower resolution. The experimental design used by Meiklejohn *et al.* (2003) , using approximately three hybridizations for every node, and the one we used in our study resulted with a GEL₅₀ of 1.64-fold in the Meiklejohn *et al.* (2003) and 1.53-fold in our study.

With a 0.001 significance threshold a GEL₅₀ of 2.65 was reached.

In addition to the GEL₅₀ we calculated the MSD₅₀ that is the median fold-change of significant expression differences, which resulted in a change of 1.74-fold with a p-value cutoff of 0.001 and a fold-change of 1.45 at the 0.05 significance level.

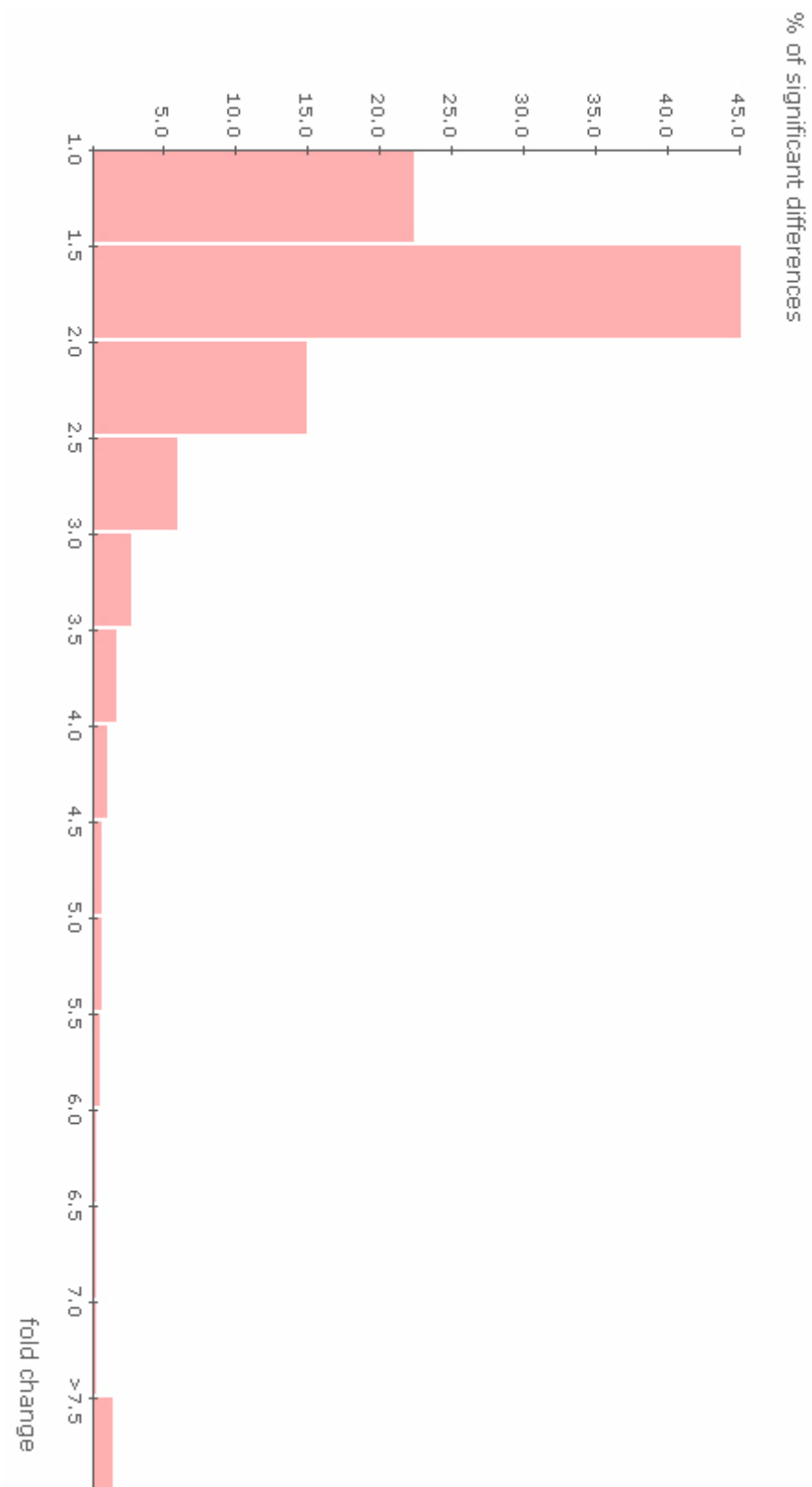


Figure 4.2 Factor of gene expression difference for the comparison of all lines ($P < 0.001$)

4.2. Inter-population comparison

To make assumptions about the gene expression difference between the African and European population, the normalized ratios of the arrays were used where an African line was compared to a European line (20 hybridizations in total). A BAGEL input file was created that contains the normalized ratios for each pairwise comparison. Each African line was named “Afro” and each European line was named “Euro”. This resulted in BAGEL estimates for two experimental nodes, where all African lines respectively all European lines were combined in a single node. Again only expressed spots were included and non-expressed spots were removed. As this approach has ten times as many hybridizations as nodes, even two expression nodes and 20 hybridizations, it should be very powerful in detecting differences in gene expression. The different hybridizations done for each node can be seen as biological replicates and therefore increase measurement precision and the probability that two samples with a given difference in gene expression level will be called significantly differentially expressed by a statistical analysis (Meiklejohn and Townsend 2005).

Also a randomly permuted data set was created by randomizing within each gene, having the same structure and proportion of missing data as the real data matrix.

	Afro	Euro
Afro		9
Euro	119	

Table 4.7 Number of significantly differentially expressed genes for inter-population comparisons ($P < 0.001$). Below diagonal: number of the real data. Above diagonal: number of the randomized data

	Afro	Euro
Afro		52
Euro	67	

Table 4.8 Number of genes with significantly higher expression in Europe given above diagonal and with significantly lower expression in Europe given below diagonal. ($P < 0.001$).

Using a significance threshold of 0.001 we obtained 119 probes out of the 9395 ones that are significantly differentially expressed between the European and Zimbabwean population as opposed to 9 probes in the randomized data set (see Table 2.7). This results in a FDR of 7.56%.

As shown in Table 4.8, 67 genes are significantly lower expressed in the European population, whereas 52 genes have a higher expression than in the African population. This leads to 79% of the genes, with a significant comparison, being higher expressed in the African than in the European population. Figure 4.4 shows the fold-changes for significant differences in the inter-population comparison. Most significant differences have a fold-change between 1.0 and 2.0.

72.27% have a fold-change between 1.0 and 1.5. As listed in Table 4.9 and shown in Figure 4.3 most of the genes which are significantly differentially expressed between Europe and Africa (21.85%) have an expression difference of approximately 1.3-fold. The biggest fold-change is 5.36 and the smallest one 1.08

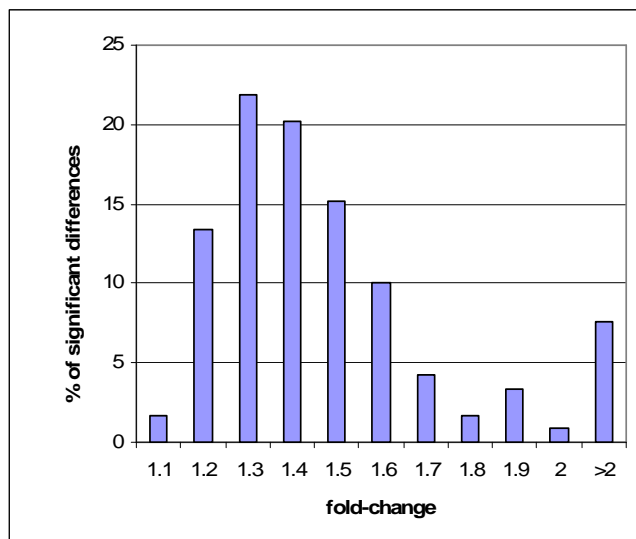


Figure 4.3 Histogram – fold change

fold-change	Frequency	%
1.1	2	1.68067227
1.2	16	13.4453782
1.3	26	21.8487395
1.4	24	20.1680672
1.5	18	15.1260504
1.6	12	10.0840336
1.7	5	4.20168067
1.8	2	1.68067227
1.9	4	3.36134454
2	1	0.84033613
>2	9	7.56302521

Table 4.9 Summary – fold change

As we did for the comparison of individual lines we also calculated the GEL_{50} and MSD_{50} for the inter-population comparison.

We obtained a GEL_{50} of 1.41 at the significance threshold 0.05 and a GEL_{50} of 2.12 at the threshold 0.001. As compared to the previous study which included all individual lines, the GEL_{50} improved, as expected, because this experimental design is much more sensitive with having ten times as many hybridizations as nodes, compared to the other one where the ratio hybridizations to nodes is 5 to 1 (see Table 4.10).

The significant median fold-change (MSD_{50}) is 1.22 for the 5% significance threshold and 1.35 with a cutoff of 0.001, and has also improved.

	comparison of lines	comparison of populations
GEL_{50} (P < 0.05)	1.53	1.41
MSD_{50} (P < 0.05)	1.45	1.22
GEL_{50} (P < 0.001)	2.65	2.12
MSD_{50} (P < 0.001)	1.74	1.35

Table 4.10 Experimental power of the two experiments

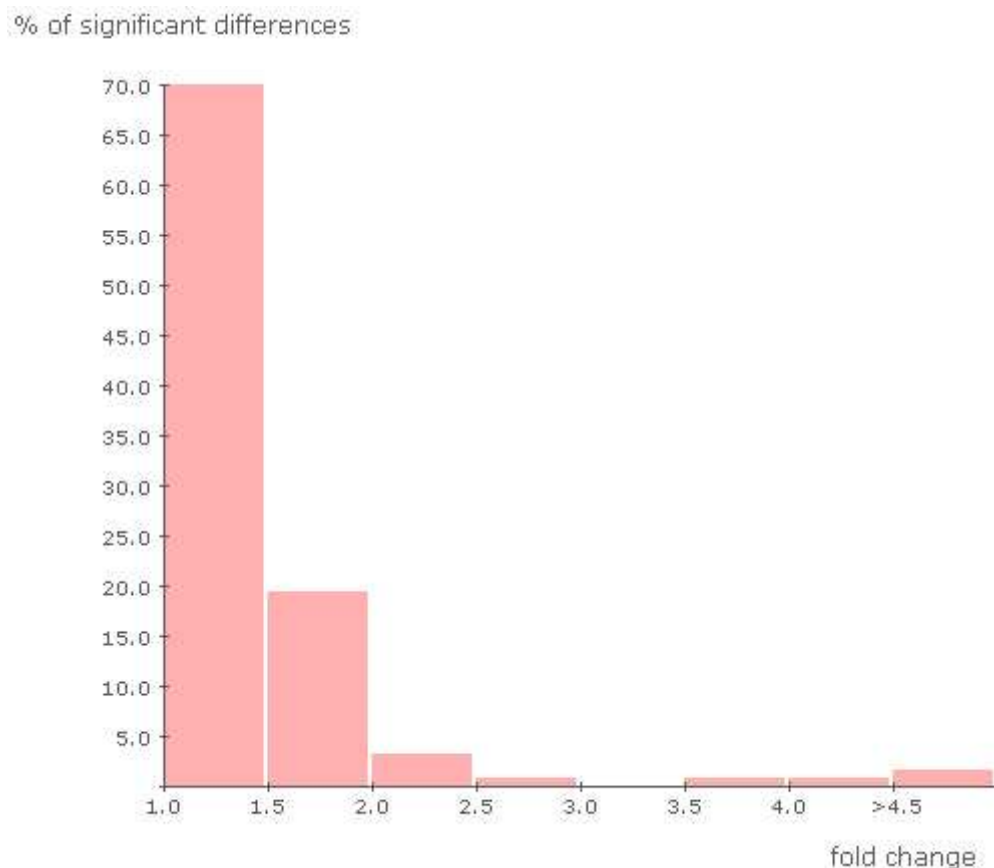


Figure 4.4 Factor of gene expression difference for the inter-population comparison ($P < 0.001$)

4.3. Expression tree

To group lines with similar expression pattern, the continuous expression levels for every line were converted into discrete states using the “expression states” function in MuMAT. We obtained a character sequence for every single line that represents the expression levels for all genes in the respective line. These sequences were used to build a tree. The *Rank Reference* method was used instead of the *Rank Neighbor* method to build a tree, because with the *Rank Neighbor* method the maximum number of different states was four as opposed to the *Rank Reference* method that results in a maximum number of eight different states and thus provides more diverse data.

Figures 4.5, 4.6 and 4.7 show the proportion of genes plotted against numbers of different states per gene at three significance levels for the randomized as well as the real data. In all three cases the real data set shows more variation per gene than it would be expected randomly. A p-value cutoff of 0.001 seems a little bit too conservative, as the maximum number of different states observed in a gene is four as opposed to seven with a p-value cutoff of 0.05.

Therefore we decided to take a significance threshold of 0.05 for this analysis, as this gives more diverse data.

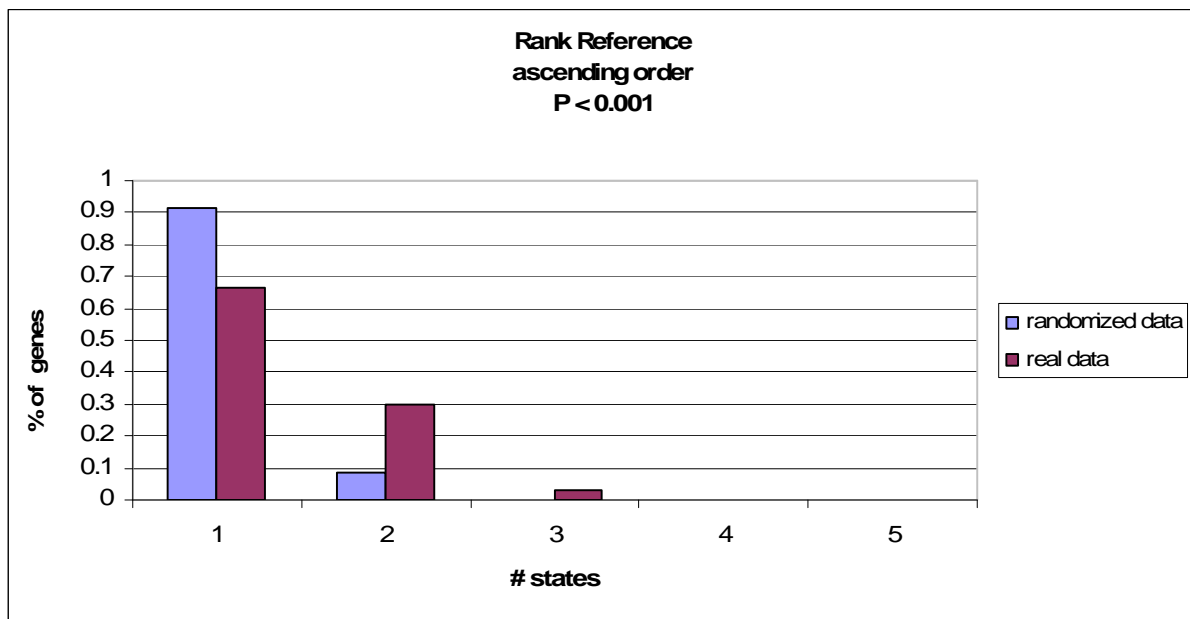


Figure 4.5 Histogram of the number of different states for all expressed genes at the $P < 0.001$ level using the Rank Reference Method with ascending order

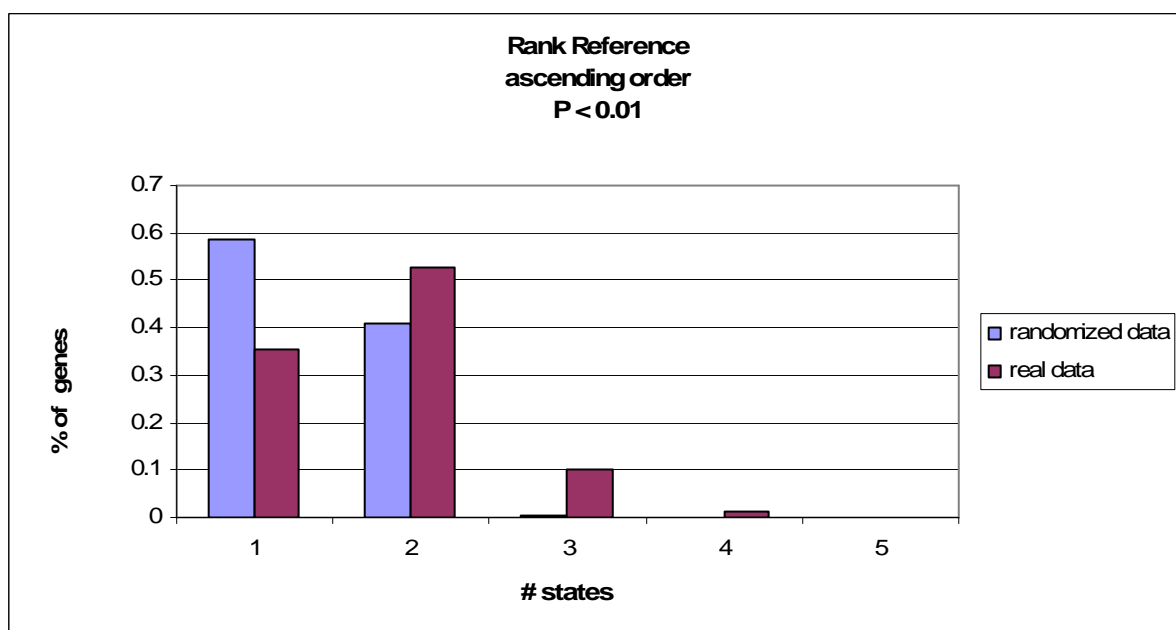


Figure 4.6 Histogram of the number of different states for all expressed genes at the $P < 0.01$ level using the Rank Reference Method with ascending order

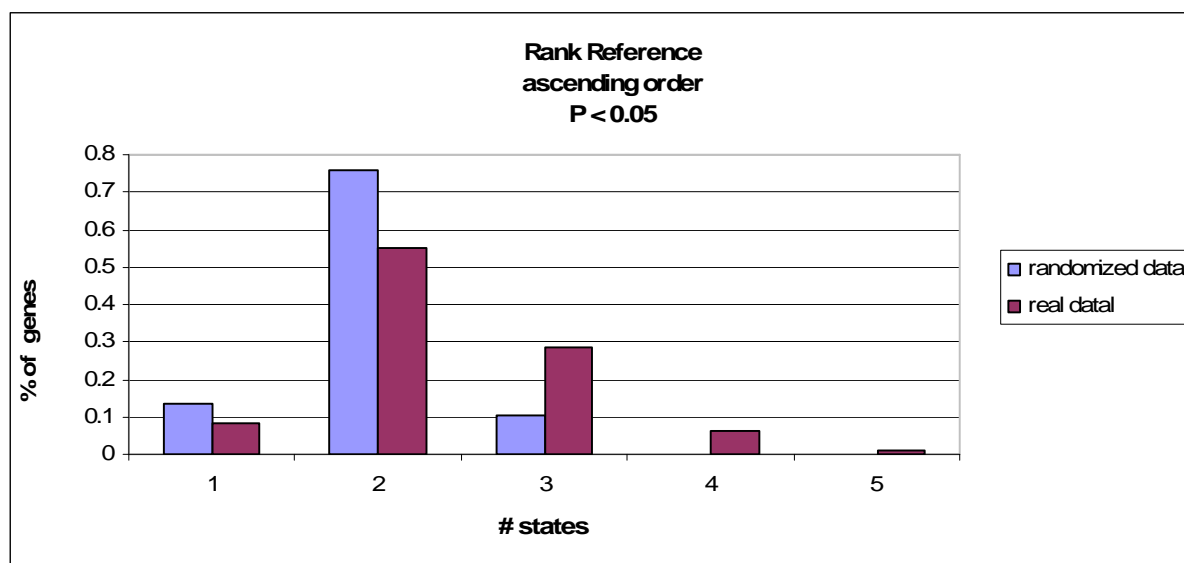


Figure 4.7 Histogram of the number of different states for all expressed genes at the $P < 0.05$ level using the Rank Reference Method with ascending order

Using PAUP* a parsimony tree was created using a heuristic search. Using the ascending order and the descending order of the Rank Reference method respectively leads to different unrooted expression trees (see Figure 4.8 and 4.9).

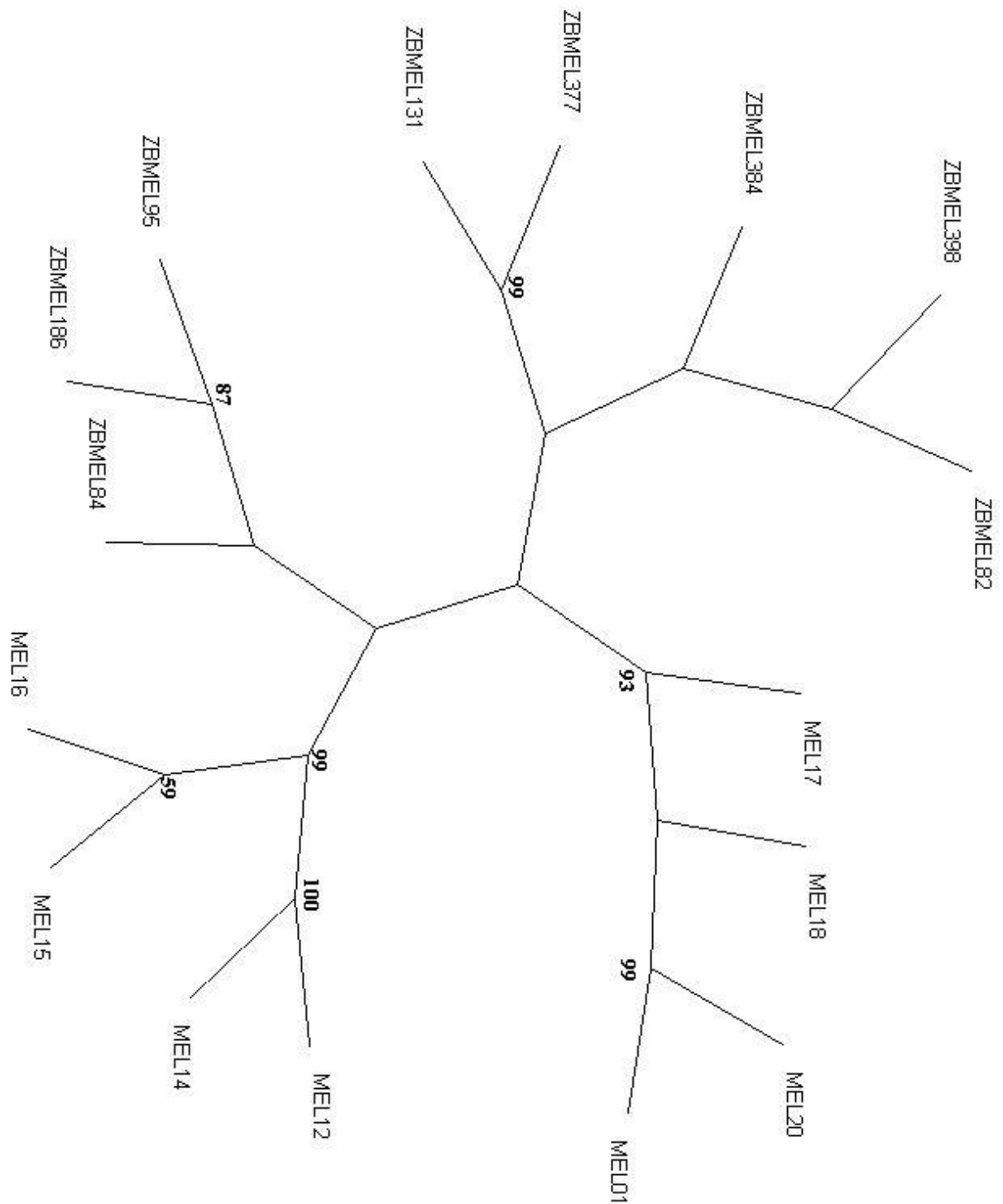


Figure 4.8 Unrooted Parsimony tree using Rank Reference method with ascending order, node labels show the confidences for the respective group estimated by a bootstrap performance

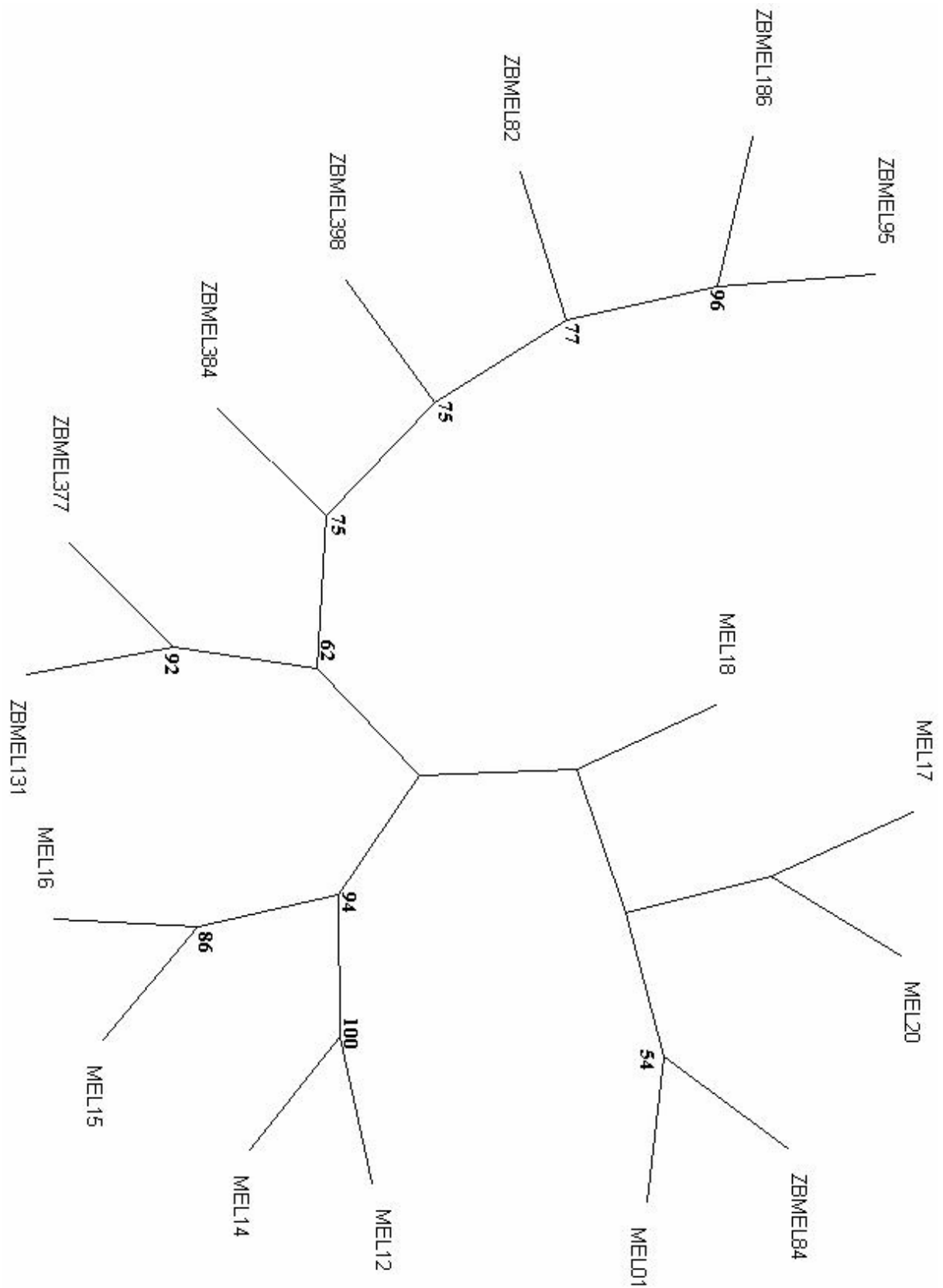


Figure 4.9 Unrooted Parsimony tree using Rank Reference method with descending order, node labels show the confidences for the respective group estimated by a bootstrap performance

The node labels indicate the percentage of the bootstrap replications that support the group descending from that branch. Only nodes are labeled where a confidence level above 50% could be found.

It can be seen that several groups are grouped together in both approaches. The confidence for the European group with the lines MEL12, MEL14, MEL15 and MEL16 is almost 100% in both trees. Also the African lines ZBMEL377, ZBMEL131 and ZBMEL186, ZBMEL95 are grouped together at a very high confidence, respectively.

The group MEL17, MEL18, MEL20 and MEL01 is also detected in both trees, but with different support. Whereas this group has a confidence below 50% in the descending ordered approach in the ascending ordered approach it is supported by a very high bootstrap value.

There is only one outlier, the African line ZBMEL84 that is associated to completely different groups in either of the trees. It is surprising that this line is closely related to MEL01 according to Figure 4.8, as comparing these lines they have the highest number of differentially expressed genes, as opposed to ZBMEL84 compared to any other line (see Table 4.2).

5. Conclusion

This thesis introduces new functions that were added to MuMAT, Munich Microarray Analysis Tool. Once the microarray experiments are completed, the data is checked, in order to remove bad quality data, the expression ratios are normalized, the background correction is performed and a BAGEL run is done to obtain relative expression levels, the data has to be analyzed. The new tools facilitate the researcher to get insight into expression difference of biological samples of interest. As in statistical inference minimizing the number of false positives is a crucial concern MuMAT provides methods to control the FDR. There is often concern for comparing the statistical power of the experimental design to other works. The GEL_{50} provides such a measurement of statistical power to detect differences in gene expression. It is shown that an experimental design that includes at least three times as many hybridizations as expression nodes will lead to higher resolution in detecting significant expression differences. Moreover tools that enable researchers to make assumptions about relationships among the samples on the basis of gene expression differences are provided.

Using the tools in MuMAT a downstream analysis was performed to compare gene expression variation of two natural populations of *Drosophila melanogaster*. Eight strains of the European as well as the African population allowed inference of gene expression variation between the different strains as well as between the two populations.

Due to the high number of hybridizations for each expression node even small differences in gene expression could be identified. The minimum fold-change detected was 1.11-fold. When examining the average number of genes that are significantly differentially expressed in Europe (126.46) and Africa (125.86) there is obviously almost no observable difference. When comparing the different lines it can be seen that the African line A95 compared to the European line E20 has most significant differences, even 322.

For most expression levels that are significantly different between lines, the factor of gene expression difference was approximately 1.6-fold. When comparing the two populations, the majority of fold-changes was approximately 1.3-fold.

Creating an expression tree using both approaches of the “expression states” function in MuMAT, it can be pointed out that European and African lines group together at a high confidence level, respectively. In both trees created, the individual lines which are grouped together are similar. Most groups that could be found in the one tree could also be verified in the second.

At the present time MuMAT provides tools for the analysis of microarray expression data that allows the researcher to manipulate the data in several ways and to get information of the quality of the data and the experimental design, as well as statistical information.

MuMAT outputs files that can be further edited to gain additional knowledge.

In the future more functions can be added to this toolbox in order to interpret the data in other ways. Tools for the functional analysis of microarray data for example could provide an insight to biological significance of the expression patterns. The significance can be explored by promoter analysis to search for evidence of transcriptional regulation and by examining the biological roles of the genes that show different expression. It is always useful to store data in standard formats like databases. It is therefore an important consideration to provide a tool that manages the keeping of data in a database. Also removing expression ratios from the data that show no expression based on the negative controls of the arrays could be implemented to avoid a manual extra step in the future.

By combining several types of tools for the analysis of microarray data in one toolbox which can be used via a user-friendly interface gives experimenters the opportunity to interpret the data in an easy automated way.

Appendix

A) List of Figures

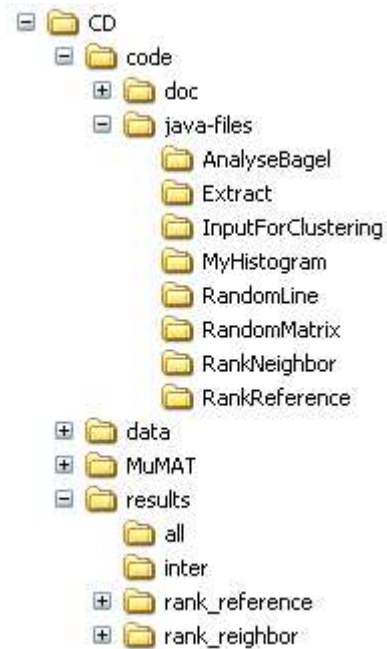
Figure 1.1	Workflow of this thesis	12
Figure 1.2	Phylogeny of the <i>Drosophila melanogaster</i> subgroup	14
Figure 1.3	Major steps in DNA microarray experiments	16
Figure 1.4	Spotted cDNA array	17
Figure 1.5	Genetix Qscan	18
Figure 2.1	Hybridizations within the populations	20
Figure 2.2	Hybridizations in the total experiment	21
Figure 2.3	Rank Neighbor	26
Figure 2.4	Rank Reference	27
Figure 2.5	Groups that were found in the bootstrap replications and their frequencies	28
Figure 3.1	MuMAT	29
Figure 3.2	Input file for BAGEL	30
Figure 3.3	BAGEL randomizer - upload	31
Figure 3.4	BAGEL randomizer – settings	31
Figure 3.5	BAGEL randomizer – download	32
Figure 3.6	BAGEL output file	33

Figure 3.7	BAGEL extractor – upload	34
Figure 3.8	BAGEL extractor – extract	34
Figure 3.9	BAGEL extractor – settings	35
Figure 3.10	BAGEL extractor – download	35
Figure 3.11	BAGEL summarizer – upload	37
Figure 3.12	BAGEL summarizer – settings	37
Figure 3.13	BAGEL summarizer – output	38
Figure 3.14	BAGEL summarizer – table of significant genes	39
Figure 3.15	BAGEL summarizer – fold change summary	40
Figure 3.16	BAGEL summarizer – histogram	41
Figure 3.17	Discrete expression states – upload	42
Figure 3.18	Discrete expression states – settings	43
Figure 3.19	Discrete expression states – output	45
Figure 4.1	Number of significant pairwise comparisons for all genes	52
Figure 4.2	Factor of gene expression difference for the comparison of all lines	54
Figure 4.3	Histogram – fold change	56
Figure 4.4	Factor of gene expression difference for the inter-population comparison	57
Figure 4.5	Histogram of the number of different states for all expressed genes (Rank Reference Method with ascending order, $P < 0.001$)	58
Figure 4.6	Histogram of the number of different states for all expressed genes (Rank Reference Method with ascending order, $P < 0.01$)	58
Figure 4.7	Histogram of the number of different states for all expressed genes (Rank Reference Method with ascending order, $P < 0.05$)	59
Figure 4.8	Unrooted Parsimony tree using Rank Reference method with ascending order	60
Figure 4.9	Unrooted Parsimony tree using Rank Reference method with descending order	61

B) List of Tables

Table 4.1	FDR at four significance thresholds	46
Table 4.2	Numbers of differentially expressed genes for each pairwise comparison (P < 0.001)	48
Table 4.3	FDR for each pairwise comparison (P < 0.001)	49
Table 4.4	Number of genes with significantly higher expression in one of the two lines (P < 0.001)	50
Table 4.5	Up/down ratio for each pairwise comparison (P < 0.001)	51
Table 4.6	GEL50 of four different citations (P < 0.05)	52
Table 4.7	Number of significantly differentially expressed genes for inter-population comparisons (P < 0.001)	55
Table 4.8	Up/down ratio for inter-population comparison (P < 0.001)	55
Table 4.9	Summary – fold change	56
Table 4.10	Experimental power of the two experiments	56

C) Content of CD



Content of CD

The source code together with a documentation of all tools that were implemented is contained on the CD. Furthermore all the data which was used for analyses are contained in the folder data. The results of the analysis described in section 4.0 are available in the folder results.

The folder MuMAT contains all files of the MuMAT toolbox.

The interface was implemented using JAVA, HTML, Perl and CGI.

Acknowledgements

I would like to thank Prof. Dr. John Parsch for giving me the opportunity to write this diploma thesis and for helping me in every aspect.

I also thank Stephan Hutter and Sarah Saminadin-Peter for making the experimental data available for me.

Bibliography

- Adams, M. D., S. E. Celniker, et al. (2000). "The genome sequence of *Drosophila melanogaster*." Science **287**(5461): 2185-95.
- Brem, R. B., G. Yvert, et al. (2002). "Genetic dissection of transcriptional regulation in budding yeast." Science **296**(5568): 752-5.
- Caccone, A., E. N. Moriyama, J. M. Gleason, L. Nigro and J. R. Powell (1996). "A Molecular Phylogeny for the *Drosophila melanogaster* Subgroup and the Problem of Polymorphism Data." Mol. Biol. Evol. **13**(9): 1224-1232.
- Clark, T. A., J.P. Townsend (2007). "Quantifying variation in gene expression" Molecular Ecology.
- Eisen, M. B., P. T. Spellman, et al. (1998). "Cluster analysis and display of genome-wide expression patterns." Proc Natl Acad Sci U S A **95**(25): 14863-8.
- Frank, I. H. W. a. E., Ed. (2005). *Data Mining: Practical machine learning tools and techniques*, 2nd Edition Morgan Kaufmann.
- Meiklejohn, C. D. and J. P. Townsend (2005). "A Bayesian method for analysing spotted microarray data." Brief Bioinform **6**(4): 318-30.
- Mount, D. W., Ed. (2004). *Bioinformatics. Sequence and Genome Analysis*, 2nd Edition, Cold Spring Harbor Laboratory Press.
- Pawitan, Y., S. Michiels, et al. (2005). "False discovery rate, sensitivity and sample size for microarray studies." Bioinformatics **21**(13): 3017-24.
- Powell, J. R. (1997). Progress and prospects in evolutionary biology : the *Drosophila* model. New York ; Oxford, Oxford University Press.
- Ranz, J. M., K. Namgyal, et al. (2004). "Anomalies in the expression profile of interspecific hybrids of *Drosophila melanogaster* and *Drosophila simulans*." Genome Res **14**(3): 373-9.
- Townsend, J. P. (2003). "Multifactorial experimental design and the transitivity of ratios with spotted DNA microarrays." BMC Genomics **4**(1): 41.

- Townsend, J. P. (2004). "Resolution of large and small differences in gene expression using models for the Bayesian analysis of gene expression levels and spotted DNA microarray." BMC Bioinformatics **5**.
- Townsend, J. P., D. Cavalieri, et al. (2003). "Population genetic variation in genome-wide gene expression." Mol Biol Evol **20**(6): 955-963.
- Townsend, J. P. and D. L. Hartl (2002). "Bayesian analysis of gene expression levels: statistical quantification of relative mRNA level across multiple strains or treatments." Genome Biol **3**(12): RESEARCH0071.
- Vinciotti, V., R. Khanin, et al. (2005). "An experimental evaluation of a loop versus a reference design for two-channel microarrays." Bioinformatics **21**(4): 492-501.