ANDROMEDA: A MATLAB Automated cDNA Microarray Data Analysis Platform

Aristotelis Chatziioannou[1], Panagiotis Moulos[1]
1 National Hellenic Research Foundation, Institute of Biological Research and Biotechnology, Metabolic Engineering and Bioinformatics Group 48 Vassileos Constantinou ave., 11635 Athens, Greece
{achatzi,pmoulos}@eie.gr
WWW home page:
http://www.eie.gr/nhrf/institutes/ibrb/programmes/metabolicengineering-en.html

Abstract. DNA microarrays constitute a relatively new biological technology which allows gene expression profiling at a global level by measuring mRNA abundance. However, the grand complexity characterizing a microarray experiment entails the development of computationally powerful tools apt for probing the biological problem studied. ANDROMEDA (Automated aND RObust Microarray Experiment Data Analysis) is a MATLAB implemented program which performs all steps of typical microarray data analysis including noise filtering processes, background correction, data normalization, statistical selection of differentially expressed genes based on parametric or non parametric statistics and hierarchical cluster analysis resulting in detailed lists of differentially expressed genes and formed clusters through a strictly defined automated workflow. Along with the completely automated procedure, ANDROMEDA offers a variety of visualization options (MA plots, boxplots, clustering images etc). Emphasis is given to the output data format which contains a substantial amount of useful information and can be easily imported in a spreadsheet supporting software or incorporated in a relational database for further processing and data mining.

1 Introduction

Functional genomics includes the analysis of large datasets derived from various biological experiments. One such type of large-scale experiment involves monitoring the expression levels of thousands of genes simultaneously under a particular condition [1] which has turned out to be a major tool for discovery in biological research. cDNA microarrays constitute a promising high-throughput technology which has become one of the indispensable tools for the inspection of genome-wide

Please use the following format when citing this chapter:

Chatziioannou, A, Moulos, P., 2007, in IFIP International Federation for Information Processing, Volume 247, Artificial Intelligence and Innovations 2007: From Theory to Applications, eds. Boukis, C., Pnevmatikakis, L., Polymenakos, L., (Boston: Springer), pp. 127-136.

changes in gene expression in an organism. Two of the frequent goals of genome-scale gene expression experiments are to identify significant alterations in transcript levels resulting from the exposure of a living system to a test agent at a given dose and time [2] and develop genetic signatures in order to distinguish between health and disease states. Additionally, such high-throughput expression profiling can be used to compare the level of gene transcription in clinical studies conditions in order to: i) identify and categorize diagnostic or prognostic biomarkers ii) classify diseases, e.g. tumours with different prognosis that are indistinguishable by microscopic examination iii) monitor the response to therapy and iv) understand the mechanisms involved in the genesis of disease processes [3].

The key physicochemical process involved in microarrays is DNA hybridization. mRNA is extracted from tissues or cells, reversed-transcribed, labelled with a fluorescent dye, usually Cy3 (green) for the reference sample and Cy5 (red) for the treated sample, and hybridized on the array using an experimental strategy that permits expression to be assayed and compared between appropriate sample pairs. Hybridization and washes are carried out under high stringency conditions to diminish the possibility of cross-hybridization between similar genes. The next step is to generate an image using laser-induced fluorescent imaging. The principle behind the quantification of expression levels is that the amount of fluorescence measured at each sequence specific location is directly proportional to the amount of mRNA with complementary sequence present in the sample analyzed. These images must then be analyzed to locate the arrayed spots and to quantify the relative fluorescence intensities for each element. Even though these experiments do not provide data on the absolute level of expression of a particular gene, they are useful to compare the expression level among conditions and genes (e.g. health vs disease, treated vs untreated) [3, 4].

The ensuing images are used to create a dataset which needs to be preprocessed prior to the subsequent analysis and interpretation of the results. Typical preprocessing steps are background noise correction, filtering procedures to eliminate non-informative genes, the calculation of the logarithmic transformed ratio between Cy5 and Cy3 channels and data normalization. The background correction is intended to adjust for non-specific hybridization such as hybridization of sample transcripts whose sequences do not perfectly match those of the probes on the array [3] and for other systematic or technical issues such as possible artefacts on the arrays, scanner setbacks, washing issues or quantum fluctuations. Other options for background correction constitute for example of using computational techniques that model the distribution of the observed intensity values and estimate the background noise based on mathematical models. The filtering procedures aim at excluding problematic or low in information content array spots and are usually based in processes that use the amount of spot noise or outlier detection to remove genes from further analysis. The ratio between channels of treated and reference samples is a simple measure which can investigate relationships between related biological samples based on expression patterns. Particularly, the $\log_2$ ratio transformation has the advantage of treating expression ratios symmetrically [4]. As another preprocessing step, normalization is considered critical to compensate for systematic differences among genes or arrays and provide appropriate balances in order to derive meaningful biological comparisons. Several reasons for normalization include unequal quantities of RNA, differences in labelling or the fluorescent dyes and systematic biases in the measured expression levels [4]. Typical normalization methods are global mean or median normalization [5], rank invariant normalization [6] and LOWESS/LOESS methods [7]. There exist several methods for normalizing cDNA microarray data and abundant literature is available on the subject [8-13].

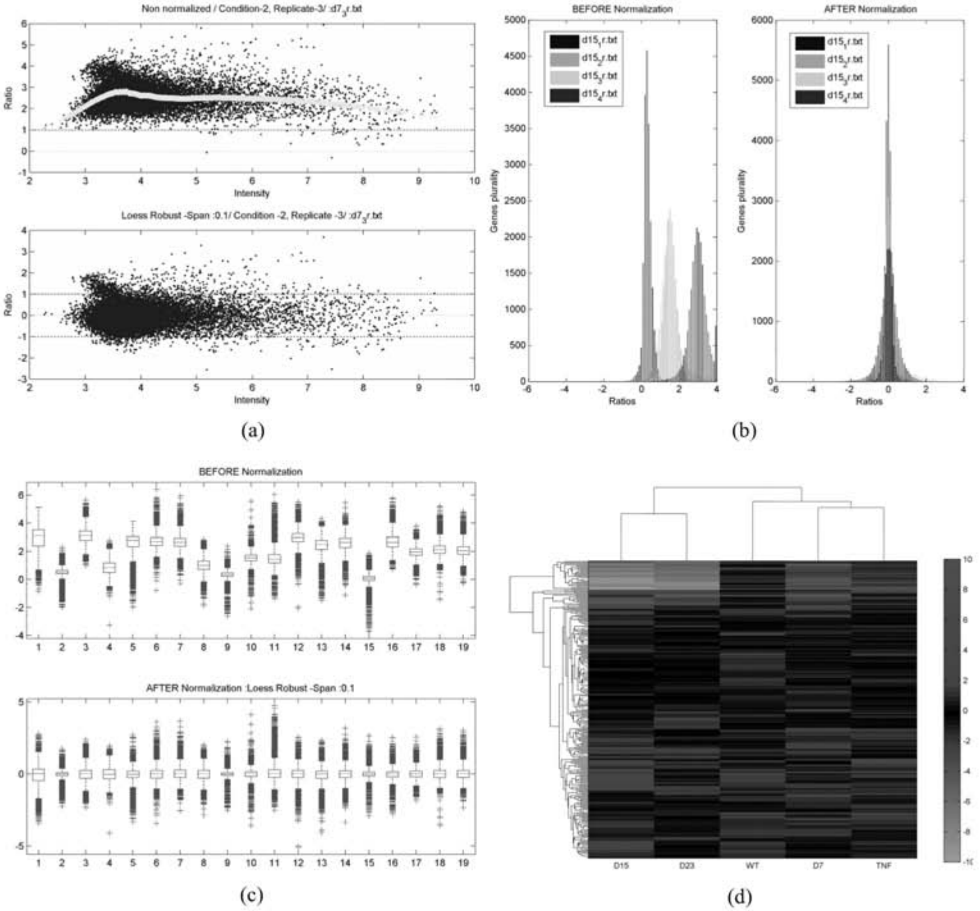


Fig. 1. (a) Example of ratio-intensity plot before (upper panel) and after (lower panel) the application of robust loess normalization with neighbouring span 0.1. The light solid line on the upper panel depicts the normalization curve. These plots display the log ratio (see section 2) for each element on the array as a function of the $\log_2(S_{Cy3}*S_{Cy5})$ product intensities and can reveal systematic intensity dependent effects in the measured log ratio values (b) Example of log ratio distributions for an experimental condition with 4 array replicates. The impact of normalization is profound (c) Example of boxplots before and after the application or robust loess normalization for a set of 19 slides. Boxplots are ideal for visualizing the result of the normalization procedure and the data spread before and after normalization (d) Example of a heatmap generated after hierarchical clustering with average linkage and euclidean distance for a set of 259 DE genes among 5 experimental conditions. Heatmaps are helpful in visualizing distinct gene clusters as well as optically distinguishing different gene groups.

There currently exist several commercial or open source microarray data analysis software packages. However, most of them are either closed black-box tools (e.g. GeneSight™, BioDiscovery Inc., Los Angeles, USA), or completely open architecture [14]. Concerning the open source solutions for microarray data analysis, although a number of software tools have been developed [15-19], a major drawback with most of them is the absence of a predefined analysis protocol leading to a batch process that commences from raw image analysis data and results in sound lists of differentially expressed (DE) genes. An exception to this rule is MIDAS software [20] where the analyst is given the ability to pre-program the analysis steps in a form of a batch procedure and caGEDA [21] which is a web-based analysis tool. Moreover, many of these analysis packages provide only several sets of routines often being of little avail to biologists with small experience in programming or scripting and other software packages which come with a graphical user interface (GUI) lack the ability to read raw data immediately after the image analysis step without certain manual transformation first [22].

The ANDROMEDA pipeline comes to fill these gaps by providing a unified environment implemented in MATLAB to form an analysis batch process, starting from reading raw image analysis software output data and resulting in lists of differentially expressed genes and gene clusters. The program utilizes a set of well defined and widely used gene filtering, normalization, and parametric or non-parametric statistical tests to analyze any number of experimental conditions and replicates and hierarchically cluster the results. Moreover, the pipeline implements additional features such as the *Trust Factor* filtering, condition based imputation of missing values and background correction based on the signal-to-noise ratio of image array spots further discussed in the Features section.

2 Features

ANDROMEDA consists of a user friendly GUI which offers a variety of options to enable different combinations among the analysis steps of a microarray experiment. It is worth noting that while most tools specialize either in data visualization and normalization, or statistical testing and clustering, ANDROMEDA offers the ability to implement all the above in a pre-defined workflow with minimal user interaction through a batch programming plug-in while being capable at the same time of performing the analysis step by step depending on the preferences of the user. Starting from the result files of the image analysis software (GenePix, ImaGene, QuantArray and text tab-delimited file formats currently supported), ANDROMEDA organizes the loaded arrays in internal data structures that will be used from the pipeline for the rest of the analysis.

After importing the data, the analysis starts by background correction to estimate the net signal for each spot, which can be performed in three different ways:

1. Background noise subtraction for each slide of each condition. In that case the net signal $\tilde{S}^{bs}$ for each channel is $\tilde{S}^{bs} = \overline{S} - \overline{B}$, where $\overline{S}$ and $\overline{B}$ are the mean or

median of the spot signal and background respectively and the log ratio between channels is:

$$R_{bs} = log_2\left(\frac{\overline{S}_{Cy5} - \overline{B}_{Cy5}}{\overline{S}_{Cy3} - \overline{B}_{Cy3}}\right) = log_2\left(\tilde{S}^{bs}_{Cy5} / \tilde{S}^{bs}_{Cy3}\right)$$

2. Signal-to-noise ratio calculation for each slide of each condition. In that case the net signal $\tilde{S}^{s2n}$ for each channel is $\tilde{S}^{s2n} = \overline{S}/\overline{B}$ and the log ratio between channels is:

$$R_{s2n} = log_2\left(\frac{\overline{S}_{Cy5}/\overline{B}_{Cy5}}{\overline{S}_{Cy3}/\overline{B}_{Cy3}}\right) = log_2\left(\tilde{S}^{s2n}_{Cy5} / \tilde{S}^{s2n}_{Cy3}\right)$$

3. No background correction. In that case the log ratio is:

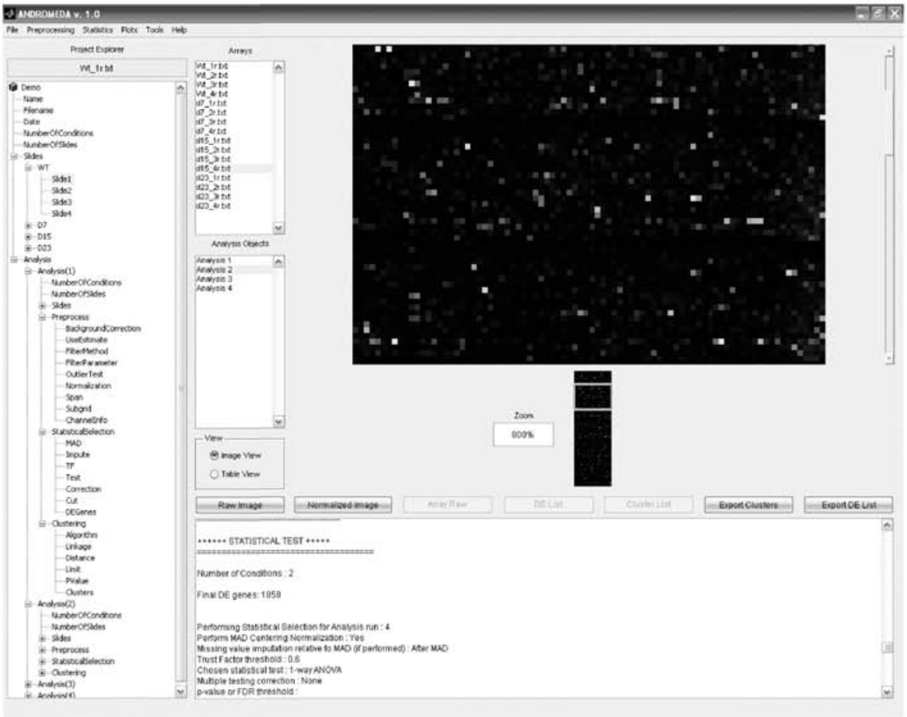
$$R_{nc} = log_2\left(\overline{S}_{Cy5} / \overline{S}_{Cy3}\right)$$

Notice that in case (1) the $\log_2$ data transformation takes place after background subtraction while in case (2) before background subtraction (immediately seen from basic properties of logarithms). Case (2) takes into consideration the signal-to-noise content of a signal, an established notion in systems theory and image processing [23], thus coming in line with the perception of the experimentalist about the quality of a signal, taking into account its interest in the strength of the signal compared to noise. This might prove critical especially when dealing with weak signal datasets whereas a majority of spot signals is close or even below the background contamination levels.

The identification of poor quality spots that will be excluded from data normalization is performed for each experimental condition in two steps: firstly spots marked as poor manually or by the image analysis software are excluded for every replicate. Then noise sensitive genes are further isolated for each slide replicate of the condition based on three possible filters according to the analyst's preference:

1. A signal-to-noise threshold filter based on the formula $\left(\overline{S}/\overline{N}\right) < BBF$ where *BBF* stands for Below Background Factor and is a user-defined threshold below which noisy spots are filtered out from the slide.
2. A filter based on the distance between the signal and the background noise distributions: a spot is robust against this filter if its signal and noise distributions abstain from each other a distance which is determined by the signal and noise standard deviations. The sensitive spots to this filter are determined by the inequality $\overline{S} - x\sigma_S^2 < \overline{B} + y\sigma_B^2$, where x and y are user-defined parameters.
3. A custom filter: in this case the investigator is given the ability to create a custom filter utilizing the signal and background means, medians and standard deviations.

In the second step, a t-test (parametric) or Wilcoxon (non-parametric) test verifies that for each spot, the ratio measurements of all condition replicates derive from a normal (or a continuous symmetrical) distribution with mean (median) equal to the average ratio for this spot among all replicates. This test tracks and excludes outliers among the replicate slides of an experimental condition. Array spots sensitive to any of these two tests are marked as non-informative poor quality spots and excluded from the estimation of the normalization curve to alleviate the normalization procedure from the impact of systematic measurement errors.



```
****** STATISTICAL TEST *****
====================================

Number of Conditions : 2

Final DE genes: 1858


Performing Statistical Selection for Analysis run : 4
Perform MAD Centering Normalization : Yes
Missing value imputation relative to MAD (if performed) : After MAD
Trust Factor threshold : 0.6
Chosen statistical test : 1-way ANOVA
Multiple testing correction : None
p-value or FDR threshold :
```

Fig. 2. A snapshot of the ANDROMEDA main window: on the left part of the interface, the project explorer allows the analyst to have a quick look at several dataset and different analyses parameters in the form of a tree. On the right next to the explorer, two lists allow the user to inspect the different arrays of the experiment as well as cycle through different analysis steps followed and display brief reports. Next to the lists on the right, each array can be viewed as an image or table depicting several values such as mean (median) signal and background intensities and their standard deviations, extracted from the image analysis software files. A message box on the bottom part of the GUI displays brief information on the steps followed throughout the analysis while several buttons offer shortcuts to different functionalities such as exporting list of DE genes. Finally, the available menus control the several functionalities as well as the variety of visualization options of the program.

ANDROMEDA continues the analysis by normalizing data on each slide separately (Figure 1a-c). If a subgrid is present on the slides, the user is given the choice to select subgrid normalization to possibly allow considering several spatial dependent properties such as local background noise in the normalization procedure. Currently, six normalization methods are supported: Global Mean, Global Median, Linear Lowess, Robust Linear Lowess, Quadratic Loess and Robust Quadratic Loess. The robust versions of Lowess/Loess [7] perform additional fitting iterations over the dataset while readjusting each point's weight on each pass, so as to mitigate the impact of possible outliers. After the normalization a subset of experimental conditions and slide replicates can be selected while the original data of the whole experiment are stored separately. With this option, the analyst may examine specific conditions or condition combinations and carry out statistical tests without having to repeat the computationally intensive and time consuming part of normalization, especially if the total number of slides is high.

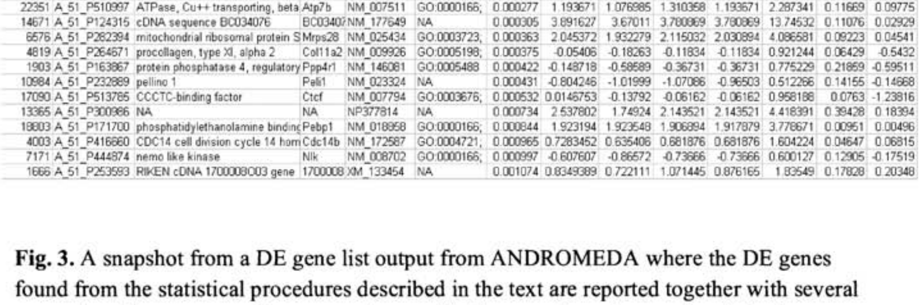
The statistical selection of DE genes starts with the calculation of the Trust Factor (TF) for each gene for all conditions and replicates and is defined for each experimental condition as: *TF = #Appearances/#Replicates*. The number of appearances for each gene is determined by the initial filtering steps: for example, if one gene in a specific slide is found sensitive either to the noise filtering condition or to the measurement reproducibility test, then it is marked as absent. If one gene is filtered out from all replicates for a given condition, then the TF for this gene is zero. This gene is then marked as 'unreliable' and is excluded from further analysis. Generally, for a biologically consistent subsequent analysis, no more than 30% of the expression values for a gene should be missing from an experimental condition (TF cutoff to 0.7). Following the TF filtering, a set of 'reliable' genes is obtained for each condition. However, since certain genes are absent from certain replicates the imputation of missing values for a gene is based on the average expression of the remaining present values of the gene of interest from the same experimental condition. Before imputing the missing values the user is given the option to perform Median Absolute Deviation [24] centering normalization to further scale the data for each condition.

The final lists of DE genes are acquired using a parametric (1-way ANOVA, [25]) or non-parametric (Kruskal-Wallis, [26]) statistical procedure, applied on the data after all the filtering and scaling steps among the subset of conditions defined after the normalization. At this point, the user has to specify certain parameters such as the statistical test to be performed or whether to correct or not for multiple statistical testing by controlling the False Discovery Rate [27] level. The final lists are formed in text tab delimited format output and they can be further manipulated effortlessly.

Finally, the DE gene lists obtained through the previous steps can be subjective to hierarchical cluster analysis, depending on the user's preference. The clustering can be performed on genes (rows), replicates (columns) or both (Figure 1d). The linkage algorithms and distance metrics supported are the single, average or complete linkage and the euclidean, standardized euclidean, Pearson correlation coefficient, Mahalanobis, Manhattan, cosine or Spearman's correlation coefficient metric respectively. The output of the cluster analysis is a clustering heatmap and lists containing formed clusters and the respective genes in MS Excel format.

3 Conclusions

The main purpose of ANDROMEDA is to provide to investigators a complete open source and flexible platform for microarray data analysis, implementing all the typical steps of the latter, starting from raw image analysis software output files up to easily interpretable and simple to manipulate gene lists. It aims both at researchers with little or no programming experience by providing a reasonably automated procedure and a user friendly interface for the analysis of microarray experiments and to those with certain expertise on statistical computation by providing completely open source routines which can be adjusted and enriched according to specific requirements. Attention is paid to the output data files which contain additional information on each spot of each slide and can be useful for drawing empirical conclusions as well as constitute key starting points for further investigation. The total procedure time of an experiment depends heavily on the amount of arrays to be analyzed as well as on several other parameters such as the normalization method (e.g. while global median normalization is preformed in less than a second, robust LOESS requires a much larger amount of time due to local data processing). Concerning the program use, apart from the graphical interface, the user can program several analysis rounds through the batch programming module. A snapshot of the program is illustrated in Figure 2 while a sample of the various outputs on Figure 3. ANDROMEDA is also provided as a library of routines which can be used individually or executed through a 'main' routine in the MATLAB command window depending on the investigator's skills.



Slide Pos	ReArrayID	Description	Symbol	GenBank	GO	p-value	26611-Cy3	26682-Cy3	27731-Cy3	Mean_Ctrl	Avg_(Not L	Std_Ctrl	CV_Ctrl
18912	A_51_P157677	PDZ and LIM domain 3	Pdlim3	NM_016798	GO:0005515;	1.43E-05	3.705595	4.112176	3.908886	3.908886	15.02076	0.20329	0.05201
20114	A_51_P104747	general transcription factor II H, p	Gtf2h4	NM_010364	GO:0003700;	1.90E-05	0.3013757	0.334892	0.318134	0.318134	1.246717	0.01676	0.05268
16993	A_51_P201945	RIKEN cDNA 4933411K20 gene	4933411	NM_025747	NA	4.25E-05	0.0149991	0.076109	0.045554	0.045554	1.032079	0.03055	0.67074
16008	A_51_P380401	RIKEN cDNA 2310051N18 gene	2310051	AK009937	GO:0004812	6.20E-05	0.5199323	0.489874	0.504903	0.504903	1.419028	0.01503	0.02977
17457	A_51_P278540	melanocyte proliferating gene 1	Myg1	NM_021713	GO:0005615	8.76E-05	0.4079994	0.495628	0.451814	0.451814	1.367759	0.04381	0.09697
14760	A_51_P212053	serine (or cysteine) peptidase inh	Serpinb1	NM_173051	GO:0000004;	8.95E-05	0.7151395	0.78945	0.752295	0.752295	1.68447	0.03716	0.04939
18330	A_51_P264495	phosphoglycerate mutase 2	Pgam2	NM_018870	GO:0003824;	0.000103	9.222468	10.18542	9.703944	9.703944	834.0235	0.48148	0.04962
6572	A_51_P318291	homeodomain interacting protein	Hipk2	NM_010433	GO:0000122;	0.000115	0.6072618	0.604532	0.601803	0.604532	1.520486	0.00273	0.00451
1588	A_51_P107998	RIKEN cDNA 9930013L23 gene	9930013	AK036809	NA	0.000124	0.8881537	0.737595	0.773229	0.799659	1.74069	0.07868	0.0984
2364	A_51_P317788	RIKEN cDNA 1110055N21 gene	Ints5	NM_176843	NA	0.000197	-0.056815	-0.15166	0.038026	-0.05681	0.961384	0.09484	-1.6693
1441	A_51_P485220	casein kinase 1, gamma 1	Csnk1g1	NM_173185	GO:0000166;	0.000215	-0.222053	-0.13621	-0.17913	-0.17913	0.883236	0.04292	-0.23962
13696	A_51_P440743	cadherin EGF LAG seven-pass G	Celsr1	NM_009886	GO:0004871;	0.000232	2.665289	3.102746	2.748885	2.838974	7.155108	0.23223	0.0818
166	A_51_P370700	glutamate oxaloacetate transamir	Got1	NM_010324	GO:0003824;	0.000239	3.436163	2.847107	3.141635	3.141635	8.825238	0.29453	0.09375
22351	A_51_P510997	ATPase, Cu++ transporting, beta	Atp7b	NM_007511	GO:0000166;	0.000277	1.193671	1.076985	1.310358	1.193671	2.287341	0.11669	0.09775
14671	A_51_P124315	cDNA sequence BC034076	BC03407	NM_177649	NA	0.000305	3.891627	3.67011	3.780869	3.780869	13.74532	0.11076	0.02929
6576	A_51_P282394	mitochondrial ribosomal protein S	Mrps28	NM_025434	GO:0003723;	0.000363	2.045372	1.932279	2.115032	2.030894	4.086581	0.09223	0.04541
4819	A_51_P264671	procollagen, type XI, alpha 2	Col11a2	NM_009926	GO:0005198;	0.000375	-0.05406	-0.18263	-0.11834	-0.11834	0.921244	0.06429	-0.5432
1903	A_51_P163867	protein phosphatase 4, regulatory	Ppp4r1	NM_146081	GO:0005488	0.000422	-0.148718	-0.58589	-0.36731	-0.36731	0.775229	0.21859	-0.59511
10984	A_51_P232889	pellino 1	Peli1	NM_023324	NA	0.000431	-0.804246	-1.01999	-1.07086	-0.96503	0.512266	0.14155	-0.14668
17090	A_51_P513785	CCCTC-binding factor	Ctcf	NM_007794	GO:0003676;	0.000532	0.0146753	-0.13792	-0.06162	-0.06162	0.958188	0.0763	-1.23816
13365	A_51_P300986	NA	NA	NP377814	NA	0.000734	2.537802	1.74924	2.143521	2.143521	4.418391	0.39428	0.18394
18803	A_51_P171700	phosphatidylethanolamine bindinc	Pebp1	NM_018858	GO:0000166;	0.000844	1.923194	1.923548	1.906894	1.917879	3.778671	0.00951	0.00496
4003	A_51_P416860	CDC14 cell division cycle 14 hom	Cdc14b	NM_172587	GO:0004721;	0.000965	0.7283452	0.635406	0.681876	0.681876	1.604224	0.04647	0.06815
7171	A_51_P444874	nemo like kinase	Nlk	NM_008702	GO:0000166;	0.000997	-0.607607	-0.86572	-0.73666	-0.73666	0.600127	0.12905	-0.17519
1666	A_51_P253593	RIKEN cDNA 1700008O03 gene	1700008	XM_133454	NA	0.001074	0.8349389	0.722111	1.071445	0.876165	1.83549	0.17828	0.20348

Fig. 3. A snapshot from a DE gene list output from ANDROMEDA where the DE genes found from the statistical procedures described in the text are reported together with several annotation elements as well as raw and normalized expression values and statistical measurements concerning the genes of each experimental condition (p-values, coefficient of variation, trust factor).

Regarding to future versions, additional features such as the support of more image analysis software formats plus model based background noise estimation will be integrated. Another goal which is currently being realized is the parallelization of the ANDROMEDA towards a powerful grid application being able to handle experiments with a vast amount of experimental conditions and large number of replicates through a simple web interface.

References

1. M. Babu, in: Computational Genomics: Theory and Applications, edited by R. Grant (Horizon Press, 2004).
2. B.A. Rosenzweig, P.S. Pine, O.E. Domon, S.M. Morris, J.J. Chen and F.D. Sistare, Dye Bias Correction in Dual-Labeled cDNA Microarray Gene Expression Measurements, *Environ Health Perspect* **112**(4), 480-487 (2004).
3. A.L. Tarca, R. Romero and S. Draghici, Analysis of Microarray Experiments of Gene Expression Profiling, *Am J Obstet Gynecol* **195**(2), 373-388 (2006).
4. J. Quackenbush, Microarray Data Normalization and Transformation, *Nat Genet* **32**, 496-501 (2002).
5. M. Bilban, L.K. Buehler, S. Head, G. Desoye and V. Quaranta, Normalizing DNA Microarray Data, *Curr Issues Mol Biol* **4**(2), 57-64 (2002).
6. G.C. Tseng, M.K. Oh, L. Rohlin, J.C. Liao and W.H. Wong, Issues in cDNA Microarray Analysis: Quality Filtering, Channel Normalization, Models of Variations and Assessment of Gene Effects, *Nucleic Acids Res* **29**(12), 2549-2557 (2001).
7. W.S. Cleveland, E. Grosse and W.M. Shyu, in: Statistical Models in S, edited by J.M. Chambers and T.J. Hastie (Wadsworth & Brooks/Cole Dormand, J.R., 1992).
8. X. Cui, M.K. Kerr and G.A. Churchill, Transformations for cDNA Microarray Data, *Stat Appl Genet Mol Biol* **2**, Article4 (2003).
9. B.P. Durbin, J.S. Hardin, D.M. Hawkins and D.M. Rocke, A Variance-Stabilizing Transformation for Gene-Expression Microarray Data, *Bioinformatics* **18**, S105-S110 (2002).
10. D.B. Finkelstein, R. Ewing, J. Gollub, F. Sterky, S. Somerville and J.M. Cherry, in: Methods of Microarray Data Analysis, edited by L. S.M. and K.F. Johnson (Kluwer Academic, Cambridge, MA, 2002) 57-68.
11. P. Hegde, R. Qi, K. Abernathy, C. Gay, S. Dharap, R. Gaspard, J.E. Hughes, E. Snesrud, N. Lee and J. Quackenbush, A Concise Guide to cDNA Microarray Analysis, *Biotechniques* **29**(3), 548-550, 552-544, 556 passim (2000).
12. J. Quackenbush, Computational Analysis of Microarray Data, *Nat Rev Genet* **2**(6), 418-427 (2001).
13. Y.H. Yang, S. Dudoit, P. Luu, D.M. Lin, V. Peng, J. Ngai and T.P. Speed, Normalization for cDNA Microarray Data: A Robust Composite Method Addressing Single and Multiple Slide Systematic Variation, *Nucleic Acids Res* **30**(4), e15 (2002).
14. R.C. Gentleman, V.J. Carey, D.M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A.J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J.Y. Yang and J. Zhang, Bioconductor: Open Software Development for Computational Biology and Bioinformatics, *Genome Biol* **5**(10), R80 (2004).
15. A. Buness, W. Huber, K. Steiner, H. Sultmann and A. Poustka, Arraymagic: Two-Colour cDNA Microarray Quality Control and Preprocessing, *Bioinformatics* **21**(4), 554-556 (2005).

16. J. Dietzsch, N. Gehlenborg and K. Nieselt, Mayday--a Microarray Data Analysis Workbench, *Bioinformatics* **22**(8), 1010-1012 (2006).
17. N. Knowlton, I.M. Dozmorov and M. Centola, Microarray Data Analysis Toolbox (Mdat): For Normalization, Adjustment and Analysis of Gene Expression Data, *Bioinformatics* **20**(18), 3687-3690 (2004).
18. R. Pieler, F. Sanchez-Cabo, H. Hackl, G.G. Thallinger and Z. Trajanoski, Arraynorm: Comprehensive Normalization and Analysis of Microarray Data, *Bioinformatics* **20**(12), 1971-1973 (2004).
19. D. Venet, Matarray: A Matlab Toolbox for Microarray Data, *Bioinformatics* **19**(5), 659-660 (2003).
20. A.I. Saeed, N.K. Bhagabati, J.C. Braisted, W. Liang, V. Sharov, E.A. Howe, J. Li, M. Thiagarajan, J.A. White and J. Quackenbush, TM4 Microarray Software Suite, *Methods Enzymol* **411**(134-193 (2006).
21. S. Patel and J. Lyons-Weiler, Cageda: A Web Application for the Integrated Analysis of Global Gene Expression Patterns in Cancer, *Appl Bioinformatics* **3**(1), 49-62 (2004).
22. A. Chatziioannou, P. Moulos, V. Aidinis and F. Kolisis, Andromeda: A Pipeline for Versatile 2-Colour cDNA Microarray Data Analysis Implemented in Matlab, (submitted to *Bioinformatics*, 2007).
23. S. de Jong and F. van der Meer *Imaging Spectrometry: Basic Principles and Prospective Applications* (Kluwer Academic, 2002)
24. J.W. Tukey *Exploratory Data Analysis* (Addison-Wesley, Reading, MA, 1977)
25. M.K. Kerr, M. Martin and G.A. Churchill, Analysis of Variance for Gene Expression Microarray Data, *J Comput Biol* 7(6), 819-837 (2000).
26. W.J. Conover *Practical Nonparametric Statistics* (Wiley, 1980).
27. Y. Benjamini and Y. Hochberg, Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing, *J R Statist Soc* **57**, 289-300 (1995).