CMST 16(1) 39-49 (2010) DOI:10.12921/cmst.2010.16.01.39-49

COMPUTATIONAL METHODS IN SCIENCE AND TECHNOLOGY **16(1)**, 39-49 (2010)

Genomic Virtual Laboratory

Barbara Cegielska[2], Damian Kaliszan[1], Luiza Handschuh[2,3]
Marek Figlerowicz[2], Norbert Meyer[1]

[1]*Poznań Supercomputing and Networking Center*
ul. Noskowskiego 10, 61-704 Poznań, Poland
[2]*Institute of Bioorganic Chemistry, Polish Academy of Sciences*
ul. Noskowskiego 12/14, 61-704 Poznań, Poland
[3]*Poznań University of Medical Sciences, Departament of Hematology*
ul. Szamarzewskiego 84, 60-569 Poznań, Poland

e-mail: vlab@man.poznan.pl

(Received: 2 February 2010; revised: 2 March 2010; accepted 10 March 2010; published online: 20 April 2010)

Abstract: In contemporary science, virtual laboratories give a chance to improve research by facilitating access to high-throughput technologies and bioinformatics methods. The Genomic Virtual Laboratory (GVL) presented here was developed for automate analysis of data retrieved from a microarray experiment. The system was implemented for R Bioconductor-based analysis of results obtained in the study on human acute myeloid leukaemia (AML). The article extends the theoretical aspects of GVL presented earlier [8] and describes how the particular elements were integrated to establish the advanced system of two-colour microarray data analysis.

Key words: genomics, microarray, virtual laboratory, remote instrumentation, measurement scenario, Bioconductor, R

I. INTRODUCTION

The Center of Excellence for Nucleic Acid-based Technologies (CENAT, IBCh PAS, Poznań) focuses on genomic and proteomic research of plant, animal and human cells. In collaboration with the Department of Haematology of the Karol Marcinkowski Medical University in Poznań, Poznań University of Technology and PSNC, the study on acute myeloid leukaemia (AML) is conducted. The project founded by the Polish Ministry of Scientific Research and Information Technology, encompasses clinical and molecular tests on bone marrow and peripheral blood samples originated from 30 patients with diagnosed acute myeloid leukaemia, subtype M1 and M2. Samples were collected in at least 3 time points (T0 – first diagnosis, T1 – post treatment, T2 – relapse or remission). The aim of the study was to identify the differences in transcriptomes and proteomes of patients and healthy volunteers, patients with various disease subtypes and samples from particular time points. The most sophisticated, expensive and time-consuming element is gene expression analysis with DNA microarrays.

DNA microarray is a tool for large-scale screening of genome and transcriptome content [9, 10]. On the solid surface, DNA probes (25-70-nt oligomers or longer PCR products), complementary to the target sequences are deposited in an ordered way [11, 12]. Then, the target molecules (genomic DNA, RNA, cDNA) are fluorescently labelled and incubated with the microarray. Labelled targets bind to the complementary probes in a process called hybrydization. As a result, fluorescence signals are detected in the positions occupied by the targets. Following laser scanning, a microarray image is quantified using specialised software. The raw data are then transformed in the normalization procedure and submitted to further analysis. The most popular microarray application is identifying aberrations in a genome sequence and gene expression profiling [13-15]. Microarrays are powerful tools that could be widely used in medical practice but have some limitations. Not every clinic can found its own microarray laboratory. Virtual laboratories can support the possibility to profit from microarray technologies without the necessity of investing in instruments and staff. A hospital which is not equipped with sophisticated devices

and does not employ specialists trained in microarray analysis can send a medical sample to a virtual laboratory where it will be processed, and then monitor the progress in microarray-based analysis. In the end the obtained results can be confronted with the other microarray data deposited in the databases.

Here, the virtual laboratory was applied for analysis of data originated from home-made, spotted microarrays that were designed for identification of genes with different expression level in acute myeloid leukaemia subtypes and stages.

II. MICROARRAY DATA ANALYSIS – EXPERIMENT DESCRIPTION

The microarray data analysis is a complex process that can be divided into the following steps (Fig. 1):

1) loading input data,
2) generating quality plots,
3) background correction,
4) inner normalization,
5) outer normalization,
6) genes extraction,
7) averaging,
8) patient filtering,
9) genes filtering,
10) clustering.

1) Loading input data

The most common programs used for quantitative analysis of microarray images (e.g. GenPix, Imagene, Agilent) generate output files in tab-delimited formats that are automatically recognized by Bioconductor. It is only necessary to use a proper command to read the particular file type. However, some files need to be transformed before reading.

VL was designed to read gpr. files, because it is the most common file format for two colour microarray raw data. Apart form the raw data file, two additional files are required: Targets file and Spot Type File.

The Targets file contains the lists of the samples that were hybridized to each array. The file can have any name and it is read into a VL session automatically. This tab-delimited text file should provide all relevant information on the experiment design. Each row corresponds with the single microarray hybridization experiment. Obligatory columns contain data input file names (that must be unique and cover with the names of the files deposited in working directory) and the IDs of samples labelled with Cy3 and Cy5 or other used pair of fluorescent dyes, e.g. Alexa 555 and Alexa 647. In classic two-color microarray experiment two samples (control and investigated) are hybridized with one microarray, each labelled with a different dye. The target file must contain the information whose dye was used for labelling each sample. Other columns are optional, serving as a source of more precise descriptions of samples, e.g. the type of cell/tissue, disease subtype, time of hybridization, notes, etc. Some of them are very useful in further analysis. The Targets file can be prepared using any text editor but spreadsheet programs such as Microsoft Excel are the most convenient.

The Spot Types file (STF) is also a tab-delimited text file and it is essential only for some tasks in the microarray data analysis procedure. It allows for identifying different types of spots from the gene list and it helps to attribute the status to each spot. STF is mainly used for distinguishing positive and negative control spots from the rest of the spots on a microarray. A SpotType column in STF encompasses the names of the different spot-types. It is important to remember that one or more other columns should correspond to the column content from the gene list, containing patterns or regular expressions sufficient to identify the spot-type. In other columns information about plotting attributes, such as colours or symbols associated with the particular spot-types can be inserted. Each row should contain a unique spot-type.

2) Generating quality plots

At the beginning of microarray data processing it is recommended to assess the quality of individual micro-arrays to decide which of the arrays should be definitely excluded from the analysis. Some of the Bioconductor packages, e.g ArrayQualityMetrics [6] offer very useful tools for diagnostic plots generation and visualization of hybridization results. Each diagnostic plot displays eight separate panels that address various aspects of array quality. These include MA-plots which help to assess intensity biases, spatial plots which can reveal local hybridization artefacts, histograms that show the signal to noise ratios for each channel, and dot plots which help to evaluate reproducibility of replicate elements.

3) Background correction

Background correction is one of the key steps in microarray data preprocessing. The part of measured probe intensities is a result of non-specific hybridization. These effects need to be removed to avoid false-positive results.

The default background correction procedure is to subtract the background intensity from the foreground intensity for each spot. However, there are many other

background correction methods which may be applied in particular situations.

In the experiments performed to identify genes with differential expression, it is preferable to do simple background subtraction to the raw data extracted from most image analysis programs. This method adjusts the foreground adaptively for the background intensities and results in strictly positive adjusted intensities, i.e. negative or zero corrected intensities are avoided. The use of an offset damps the variation of the log-ratios for very low intensities spots towards zero.

As it was mentioned, different data sets require different background correction methods. The Genomic Virtual Laboratory enables the user to choose one of methods from the following list:

a. none,
b. subtract,
c. half,
d. minimum,
e. movingmim,
f. edwards,
g. normexp.

The background correction step is mainly based on the limma package which is currently the best package for two-colour microarray data preprocessing.

Normalization

Normalization enables to compare measurements within and between arrays from separate experiments. The main goal of this step is to reduce non-specific signals resulting from not uniform washing and other technical problems that occur during the experiment execution. For example, normalization reduces such sources of variation as: different efficiency of reverse transcription, labelling, hybridization reaction, physical problems with array spotting, and reagent batch effects. Some errors can be prevented by proper microarray construction – using spot replicates in random positions across the array or using sets of control probes (spikes). These probes do not hybridize with the target sample but are complementary to the external RNA molecules, added to the reaction mixture in a predefined amount Spike controls are very helpful during normalization and they are definitely more reliable than so called housekeeping genes (genes believed to reveal stable expression independently on the cell type and developmental stage) [5].

Normalization can proceed in various ways, depending on the experiment set-up. In the presented virtual laboratory two kinds of normalization are distinguished:

- inner normalization,
- outer normalization.

4) Inner normalization

In biological experiments involving microarrays covering the whole genome/transcriptome only a relatively small number of genes is expected to be differentially expressed. The remaining genes have generally a constant expression level, comparable between control and investigated samples. It means that almost all the genes on the array can be used for normalization in order to minimalize the observed discrepancy between signal intensities detected for two channels. In other words, the majority of genes can function as indicators of the dye bias. Balancing this bias should therefore not affect the genes that vary significantly in expression between two studied samples [5].

At this stage of analysis it must be decided which set of genes is to be used for normalization. A number of considerations influence this decision, such as the proportion of genes that are expected to be differentially expressed in the studied samples, or availability of control probes and the target sequences. That is why GVL implements a range of normalization methods for spotted microarrays:

a. none,
b. median,
c. loess,
d. printtiploess,
e. composite,
f. control,
g. robustspline,
h. vsn.

The inner normalization step uses the normalizeWithinArrays function (limma package) and is one of the best functions that can be applied for normalization. Listed methods were described as the most commonly used methods for normalization of spotted arrays, especially loess and printiploess [5].

5) Outer normalization

This step supports some of the methods available for between-array normalization of two color microarrays:

a. none,
b. scale,
c. quantile,
d. Aquantile,
e. Gquantile,
f. Rquantile,
g. Tquantile.

A feature which distinguishes most of these methods from inner normalization is that they focus on the individual red and green intensity values rather than merely on the log-ratios. These methods might be called individual chan-

nel or separate channel normalization methods. An important issue to consider before outer normalization is how background correction has been handled. That is why the block sequence in each measurement scenario is so important. Outer normalization is effective when there are no missing values in log-ratios which might arise from negative or zero corrected intensities. This step was built using functions and commands from limma and vsn packages.

6) Genes extraction

From this stage the first set of biologically relevant information is extracted. Here, an approach called linear models (limma) is applied to analyse data obtained from microarray experiments. This approach treats all single experiments as a simple replicated experiment [1, 2]. Two matrices are specified. The first is the design matrix indicating which sample has been applied to each array. The second one is the contrast matrix defining comparisons that the user would like to make between the studied samples. Very simple experiments do not need to define the contrast matrix as it can be easily predicted on the base of the design matrix. The procedure starts with fitting a linear model to the user data. The model is specified by the design matrix. Each row of the design matrix should therefore correspond to a single array experiment and each column should correspond to a coefficient which is used to describe the sample origin and status in the experiment. The main purpose of this step is to assess the variability in the data and distinguish this data-specific variability from random variation. That is why the systematic part needs to be modelled properly. The limma package applies the eBayes function which computes a number of summary statistics (B-statistics, F-statistics, t-statistics, p-value, etc.) for each gene and each contrast. The p-values calculated by eBayes are not adjusted for multiple testing. Such adjustment should be done and can be achieved by different functions. It is also possible for a GVL user to choose a method of adjustment:

a. none
b. BH
c. BY
d. Holm

Prior to linear model fitting the probes summarization is introduced in this block. It is required when transcripts are represented by multiple probes (probe sets). For each gene, the background adjusted and normalized intensities for the probes representing the same target are summarized.

7) Averaging

In the previous stages of the analysis all arrays were treated as biological replicates. If the Targets file contains arrays that are technical replicates it must be taken into account because the technical replicate pairs are not independent. Actually, they are more likely to be positively correlated. During this step a vector indicating the two blocks corresponding to biological replicates is created. This model of analysis reminds the mixed model analysis of variance [4] except the fact that information is borrowed between genes. Information is borrowed by constraining the within-block correlations to be equal between genes and by using empirical Bayes methods to moderate the standard deviations between genes [3]. This step is based on the limma package.

8) Patient filtering

This block is definitely the most interesting option developed in the Genomic Virtual Laboratory. It allows the user to filter only one microarray (corresponding to a particular specimen/patient) from the whole data set. It is very useful especially when the experiment is devoted to biomedical research. For medical doctors (and for patients themselves) the most relevant is the information concerning the individual person. For instance, from the group study devoted to a particular disease (as AML) it is possible to filter only those genes that are over- or underexpressed in a sample from a single patient. Moreover, by using gene extraction the user can compare the results of gene expression obtained for a single patient with the results of the complete data set. It is a totally new approach and, what is important, it is statistically acceptable due to technical replicates of microarrays and the fact that the patient filtration occurs after inner or outer normalization (depending on user preferences). It means that the total data set is normalized equally, which highly reduces possible statistical errors.

9) Genes filtering

An average microarray contains probes identifying from hundreds to thousands of transcripts. It is commonly known that not all genes located on the microarray are transcribed in the studied samples. Some probes are undetectable and others give very weak signals. The main goal of the filtration process is to remove such probes in order to avoid false positive results. On the other hand, after comparing an investigated sample with a control one, it can be noticed that the level of gene expression is changed in even a lower number of genes located on the array. To filter genes of interest from the data set, many

filtration methods can potentially be applied. The simplest selection is based on changes in the gene expression level. It estimates the intensity ratio of the investigated sample vs. the control sample in log-scale. Genes with expression change fold higher than 1.75-2 and smaller than 0.5-0.75 are the genes of interest. The remaining genes are excluded from further analysis. This approach is sensible but not resistant to statistical errors. Type I errors generate false positive results by leaving genes with an unchanged expression level, and type II errors generate false negative results by excluding genes with differential expression. The "Genes filtering" step was designed to solve these problems. As it was mentioned before, the gene extraction step produces a number of summary statistics. P-values assess the probability of obtaining the same or better random result. The default cutting point is $p < 0.05$, whicht means that there are less than 5% of chances that the observed result would be obtained by accident. However, it means that during analysis of 20 000 genes 1000 genes can be misclassified. At this stage of analysis the user must define the cut-off point him/herself. This cut-off point can be different for different data sets, compromising the statistical accuracy and biological relevance. Filtering step was equipped with multiple testing procedures, based mainly on multtest and genefilter packages [7].

10) Clustering

Cluster analysis is the assignment of a set of observations into clusters (subsets), so that observations in the same cluster are similar in any way. User starts with the normalized microarray data that are to be subdivided into homogeneous groups. First, the variables are chosen for assessment which groups are expected to be similar. Next, the decision is made whether to standardize the variables in some way so that they all contribute equally to the distance or similarity between cases. Finally, the user decides which clustering procedure to use, basing on the number of cases and types of variables that are going to be used for forming clusters.

For hierarchical clustering, the chosen statistics quantifies how far apart (or similar) two cases are. Then the method of forming groups is selected.

In *k*-means clustering, the user selects the number of clusters. The algorithm iteratively estimates the cluster means and assigns to the cluster each case for which its distance to the cluster mean is the smallest.

Clustering with the pvclust package is used for assessing the uncertainty in the hierarchical cluster analysis. For each hierarchical cluster, quantities called *p*-values are calculated via multiscale bootstrap resampling. *P*-value of a cluster is a value between 0 and 1, indicating how strong the cluster is supported by data. What is important, pvclust provides two types of *p*-values: AU (Approximately Unbiased) *p*-value and BP (Bootstrap Probability) value. AU *p*-value, which is computed by multiscale bootstrap resampling, is a better approximation to unbiased *p*-value than BP value computed by normal bootstrap resampling.

This step was built using functions from pvclust, fpc, and mclust packages.

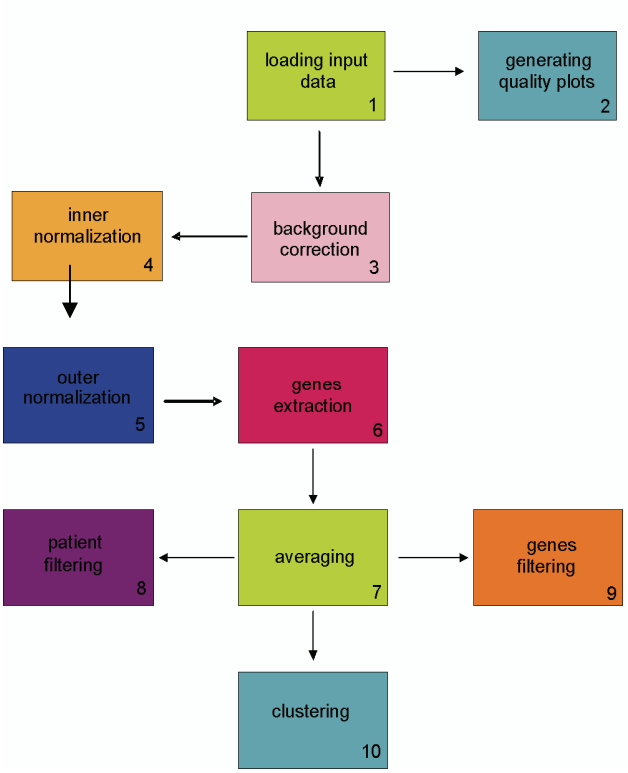


Fig. 1. Scenario diagram for genomic R scripts

III. GENOMIC VIRTUAL LABORATORY – ARCHITECTURE OF SYSTEM

Virtual Laboratory [18] has been created to develop a universal system architecture capable of handling a wide variety of different laboratory instruments (seen in the VL as resources), software controlling instruments or software embedded in the Grid environment itself (considered as system tasks).

To adapt genomic domain to the VL [8] specification the diagram described in Section II presenting R scripts had to be mapped to VL tasks.

One GVL resource represents exactly one block of the diagram. One block corresponds to one R script. In other

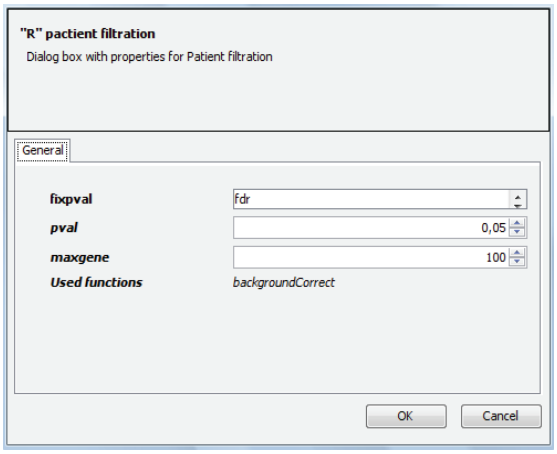


Fig. 2. Resource properties dialog window

words, the workflow contains as many stages as R scripts. ll stages can be connected between themselves only in the given order supervised by the system itself.

A script is seen in the GVL as a set of resource attributes. For implementation purposes, it has been broken down into two parts. The first one is a declaration of variables which values are passed from the Workflow Editor Application [8] – see Fig. 3. These variables can be modified in the resource properties dialog window accessed by a double-click. Additionally it provides information about specific functions used in the computations. The second, executable part is never changed. Before execution, these two parts are combined with each other and thus a ready to launch script is created.

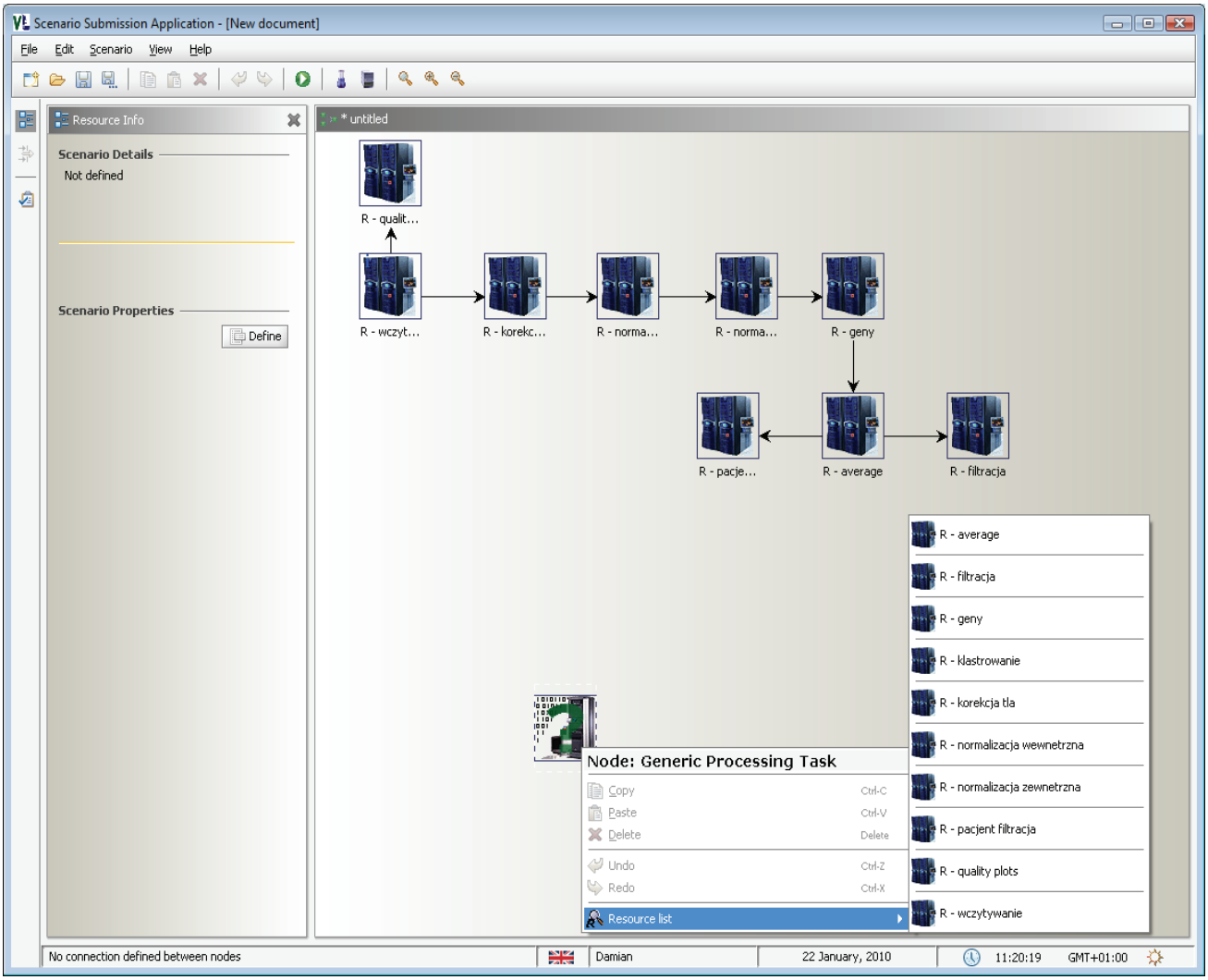


Fig. 3. Workflow Editor Application

IV. OUTPUT DATA

At the 'patient filtering' stage the list of CSV files with all processed biological samples are received. Each file contains many interesting data from which we extract genes having differential expression and their values. This information needs to be converted to XML format and then sent via a web service to the external system (which in turn updates the Eskulap subsystem) in the format given below:

```
<?xml version="1.0" encoding="UTF-8" ?>
<Opis_przypadku id="String"
xsi:noNamespaceSchemaLocation="Opis_przypadku_
0.9.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema
-instance">
<Etap data_zakonczenia="2010-01-13"
data_rozpoczecia="2009-08-13"
faza_choroby="String" etap="String">
<Badania_laboratoryjne>
        <Badanie_ekspresji_genow id_probki="String"
        id_badania="String"
        miejsce_badania="String" komentarz="text"
        rodzaj_probki="String" data_badania="1967-
        08-13" id_miejsca_badania="String">
      <Gen>
        <Nr_genu>No</Nr_genu>
      <Symbol>Symbol</Symbol>
      <Nazwa>Symbol</Nazwa>
      <Poziom_ekspresji>value</Poziom_ekspresji>

      <Rodzaj_ekspresji>expr_type</Rodzaj_ekspresj
      i>
        </Gen>
      <Gen>
        <Nr_genu>No</Nr_genu>
      <Symbol>Symbol</Symbol>
      <Nazwa>Symbol</Nazwa>
      <Poziom_ekspresji>value</Poziom_ekspresji>

      <Rodzaj_ekspresji>expr_type</Rodzaj_ekspresj
      i>
      </Gen>
      ...
    </Badanie_ekspresji_genow>
     ...
     <Badania_laboratoryjne>
</Opis_przypadku>
```

The main top-down structure of the above data is as follows:

1. Case description having the attributes:
    a. Id – randomly generated GUID [17] passed to VL system along with a patient's sample
2. Stage
    a. Finish date – the date when the given stage of a patient's treatment is finished
    b. Start date – the date when the given stage started
    c. Phase – Phase name (we distinguish the following phases: *de novo,* resistance, relapse 1, relapse 2 … relapse *n*)
    d. Stage – Stage name (e.g. diagnostics, remission induction, consolidation of remission, relapse, etc.)
3. Laboratory examination
    a. Genes expression examination – the attributes are as follows:
        i. Sample id
        ii. Examination id
        iii. Examination site name and id
        iv. Comment
        v. Sample type (blood, bone marrow)
        vi. Examination date
    b. List of genes – delivering gene number, name, symbol, expression level and type (high, low)
4. Other examinations

The result CSV file, in turn, has the following structure: the first line contains the column names: Block, Row, Column, ID, Name, logFC, AveExpr, t, P.Value, adj.P.Val, B. A The last five columns indicate which of the genes from the list are differentially expressed. The choice of the column and the cut-off point depend on the user. To facilitate the choice, genes are ranked according to adj.P.Val column.

V. DIGITAL LIBRARY

Output data provided by scripts executed in the GVL at each stage are stored in the Digital Library based on the Data Management System [16] created at PSNC and implemented in many earlier projects. The results of analysis can be re-used or shared with other scientists working in the system, if needed. They may be found helpful for the doctors treating their patients. Some important information extracted from result files may influence e.g. the patient treatment process or provide other valuable medical knowledge.

Mostly, the data coming from 'patient filtering' stage need to be exported to Eskulap – an external complex IT platform for health care installed at the hospital. In this way the data coming from the hospital as a sample go back as digitized and processed data. The information comes a full circle.

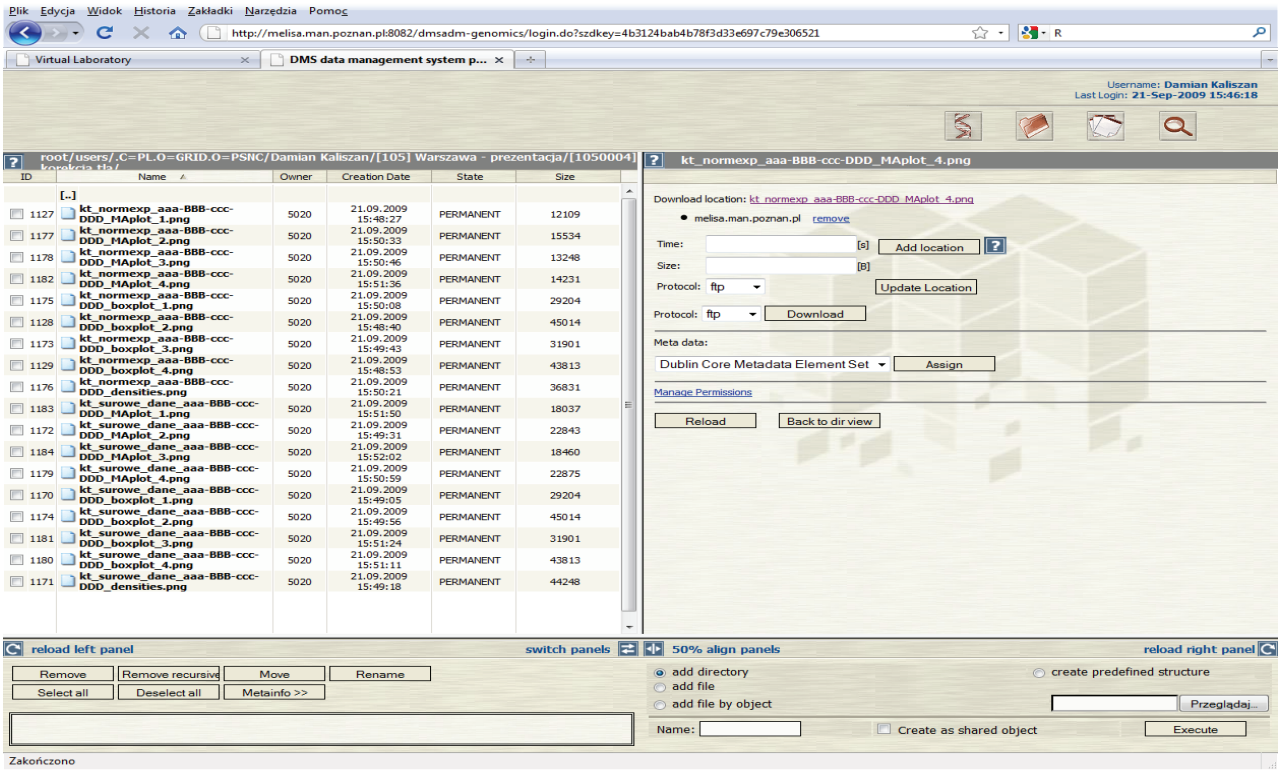


Fig. 4. Digital Library with output files

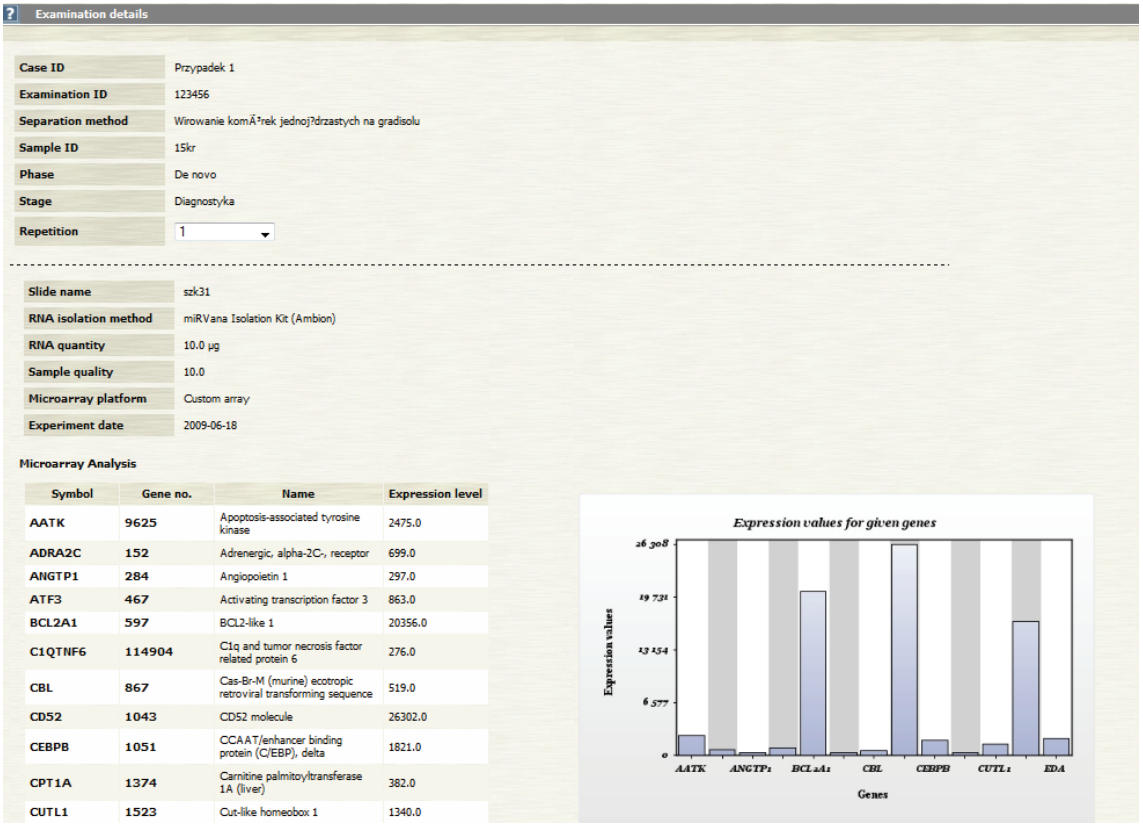
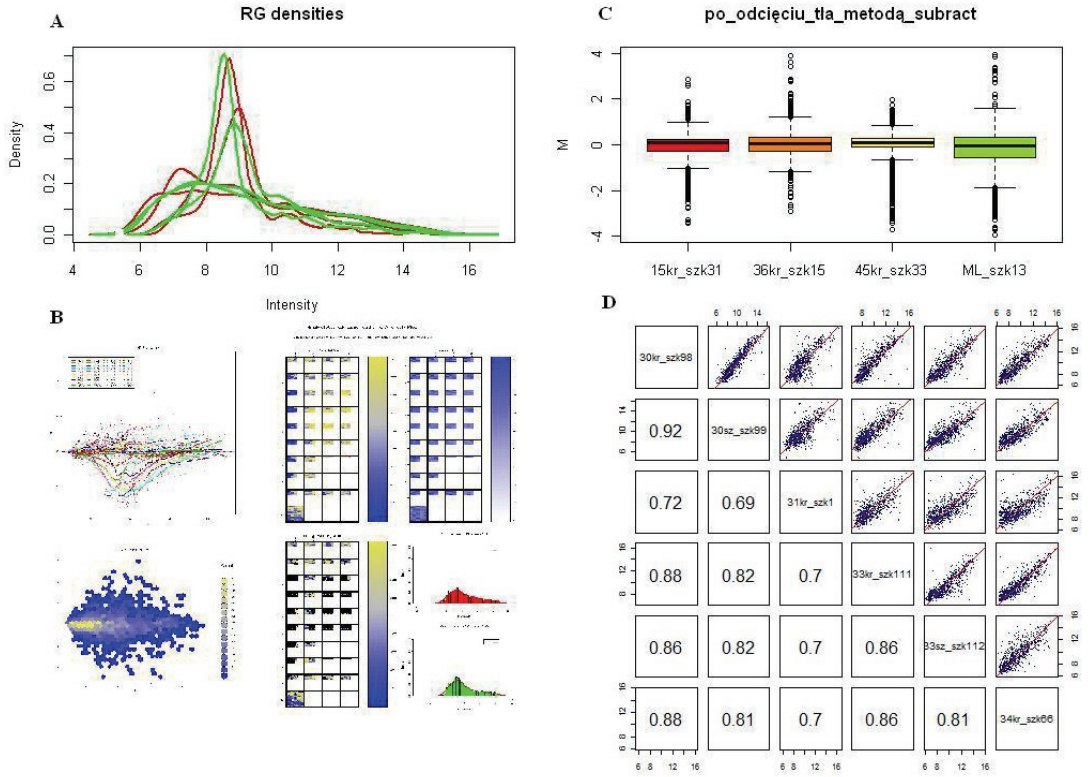


Fig. 5. Digital Library – patient details



Density plot is used to view the distribution of variables, B – Diagnostic plot, containing MA-plots, spatial plots and histograms, C - Boxplot, D – Pairwise plots

Fig. 6. Examples of graphics generated in GVL using Bioconductor-based scripts

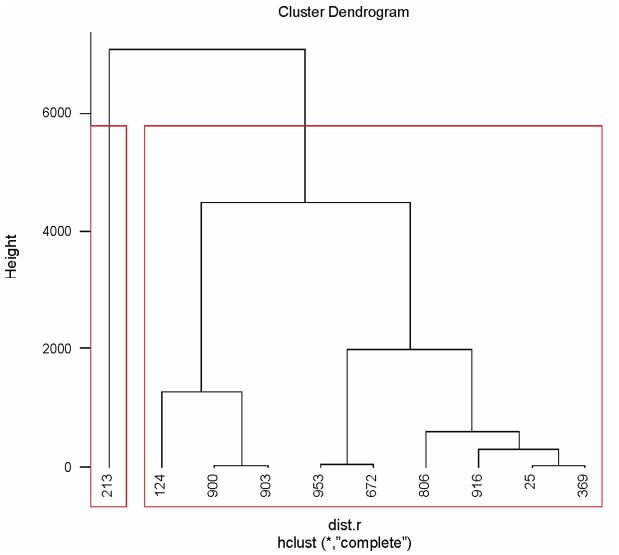


Fig. 7. The result of hierarchical clustering

Digital Library can be used in several user modes called e.g. Files and Genomics. The top-down structure of the former is as follows: user folder, scenario main folders (with the names equal to scenarios names), sub-folders (with names equals to resources names) (Fig. 4)

The latter provides the list of examined patients with stages and phases of the illness, etc. By clicking on the selected option we receive details for the patient, such as: case id, examination id, separation method, sample id, phase name, stage name, list of genes with differential expressions that are additionally presented in the bar chart (Fig. 5). The list of genes with their expressions values is stored in the XML described in Section IV.

VI. PRELIMINARY RESULTS – EXAMPLES

As the microarray experiments are still in progress, the GVL action was monitored using a test set of samples (raw data from 86 two-colour microarrays). The microarrays were analysed using all steps described in Section II. The examples of preliminary results are presented at Fig. 6 and Fig. 7.

VII. SUMMARY

Here we presented the application of the virtual laboratory system for analysis of two-colour microarray data. The system based on Bioconductor R scripts has a modular architecture. Particular blocks cover the following steps of preprocessing and high-level analysis. The raw data in tab-delimited files are introduced into the environment and submitted to background correction, normalization, filtering, clustering and differential expression analysis. Plots and output files generated during data processing are automatically saved and stored in the digital library. The most practical usage of the GVL system is a possibility to extract information about a single sample (patient) and relate it to the results obtained for the whole experimental data set.

Acknowledgments

The work was supported by the grant from the Polish State Committee for Scientific Research No. PBZ-MniI-2/1/2005 to M.F.

References

[1] G.K. Smyth, *Linear models and empirical Bayes methods for assessing differential expression in microarray experiments*. Statistical Applications in Genetics and Molecular Biology 3, Article 3, 2004.
[2] Y.H. Yang, T.P. Speed, *Design and analysis of comparative microarray experiments.* In: T.P. Speed, editor, Statistical Analysis of Gene Expression Microarray Data, pages 35-91. Chapman & Hall/CRC Press, 2003.
[3] G.K. Smyth, J. Michaud, H. Scott, *The use of within-array replicate spots for assessing differential expression in microarray experiments.* Bioinformatics 21 (9), 2067-2075 (2005).
[4] G.A. Milliken, D.E. Johnson, *Analysis of Messy Data*. Volume 1: *Designed Experiments*. Chapman & Hall, New York, 1992.
[5] Y.H. Yang, S. Dudoit, P. Luu, T.P. Speed. Normalization for cDNA Microarray Data SPIE BiOS 2001, San Jose, California, January 2001.
[6] R. Gentleman, V.J. Carey, W. Huber, R.A. Irizarry, S. Dudoit, *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Springer.
[7] F. Hahne, W. Huber, R. Gentleman, S. Falcon, *Bioconductor Case Studies*. Springer.
[8] L. Handschuh, M. Lawenda, M. Stępniak, M. Figlerowicz, M. Stroiński, J. Węglarz, Computational Methods in Science and Technology 15 (1), 31-40 (2009).
[9] V. Trevino, F. Falciani, H.A. Barrera-Saldaña, *DNA microarrays: a powerful genomic tool for biomedical and clinical research*, Mol. Med. 13, 527-541 (2007).
[10] L.A. Garraway, W.R. Sellers, *Array-based approaches to cancer genome analysis.* Drug Discov. Today 2 (2), 171-177 (2005).
[11] D.N. Howbrook, A.M. van der Valk, M.C. O'Shaugnessy, D.K. Sarker, S.C. Baker, A.W. Lloyd, *Developments in microarray technologies*. Drug Discov. Today 8 (14), 642-651 (2003)
[12] S. Venkatasubbarao, *Microarrays – status and prospects*. Trends Biotechnol. 22, 630-637 (2004).
[13] T. Haferlach, A. Kohlmann, S. Schnittger, M. Dugas, W. Hiddemann, W. Kern, C. Schoch, *Global approach to the diagnosis of leukemia using gene expression profiling*. Blood 4, 1189-1198 (2005).
[14] O. Margalit, R. Somech, N. Amariglio, G. Rechavi, *Microarray-based gene expression profiling of hematologic malignancies: basic concepts and clinical applications*. Blood Rev. 19, 223-234 (2005).
[15] X. Chen, E. Jorgenson, S.T. Cheung, *New tools for functional genomic analysis.* Drug Discov. Today 14 (15-16), 754-760 (2009).
[16] Data Management System Web page http://dms.progress.psnc.pl/
[17] http://pl.wikipedia.org/wiki/GUID
[18] http://vlab.psnc.pl

BARBARA CEGIELSKA received the M.Sc. in Biotechnology in 2008 from Wrocław University of Technology. She is currently a Ph.D student in the Center of Excellence for Nucleic Acid-Based Technologies, Institute of Bioorganic Chemistry, PAS. In 2009 she started to study bioinformatics at the Adam Mickiewicz University in Poznań.

DAMIAN KALISZAN graduated from Poznań University of Technology and received his M.Sc. in Computer Science (Computer Integrated Management and Production Systems) in 2001. He currently works in Supercomputing Department at PSNC on the Virtual Laboratory (http://vlab.psnc.pl) and other EU projects. His research interests include data mining, web and Java related technologies.

DR. LUIZA HANDSCHUH graduated in biology from the Adam Mickiewicz University in Poznań in 1998. She received her PhD degree from the Institute of Bioorganic Chemistry in 2004. Since 2005 she has been a specialist in the Department of Haematology, Poznań University of Medical Sciences, and since 2006 – a associate in the Center of Excellence for Nucleic Acid Based Technologies, one of the laboratories affiliated to the Institute of Bioorganic Chemistry in Poznań. Her main interest is functional genomics, DNA microarray technology and its application in medical science. She participates in many research projects, collaborating with other Polish laboratories. At present, she is one of the principal investigators in the project devoted to gene expression profiling in a human acute myeloid leukemia.

PROF. M. FIGLEROWICZ (MF), graduated in chemistry from the Poznań Institute of Technology in 1985. He received the PhD degree in biochemistry from the Adam Mickiewicz University in 1991 and the D.Sc (habilitation) degree in biology from the same university in 2000. In 2006 he acquired the title of professor of biology from the President of Poland. His research interests concern: RNA biology, especially small regulatory RNA, molecular biology of RNA viruses and retroviruses, functional genomics. He has been the principal investigator of numerous scientific projects sponsored by the Polish Ministry of Science and High Education, European Union and UNESCO. He is the author or co-author of 91 articles and 122 published scientific communications. As an invited speaker, he has given 47 lectures and seminars. Currently, he is the head of the Laboratory of Plant Molecular Biology at the Institute of Bioorganic Chemistry (ICHB) Polish Academy of Sciences, Poznań, the head of the Center of Excellence for Nucleic Acid Based Technologies (CENAT), located at the ICHB and the head of Laboratory of Molecular Virology at ICHB. He is also a member of several scientific boards, the editorial board of Journal of RNAi and Gene Silencing, Oxford United Kingdom and a member of the editorial board of Biotechnologia – Journal of the Biotechnology Committee of the Polish Academy of Sciences.

DR. NORBERT MEYER is currently the head of the Supercomputing Department in Poznań Supercomputing and Networking Center (http://www.man.poznan.pl). His research interests concern resource management in GRID environment, GRID accounting, data management, technology of development graphical user interfaces and network security, mainly in the aspects of connecting independent, geographically distant Grid domains. The concept of remote operation, controling and monitoring instrumentation of one of key research topics he is currently performing on national and international levels. Norbert Meyer conceived the idea of connecting Polish supercomputing centres, vision of dedicated application servers and distributed storage infrastructure. He is the author and co-author of 60+ conference papers and articles in international journals, member of programme committees of international IT conferences. Member of the eIRG group in EC (www.eirg.org).