MDAT: Microarray Data Analysis Tool Deployment as SaaS in the Cloud

Monika Simjanoska, Ana Madevska Bogdanova and Marjan Gusev
Ss. Cyril and Methodius University,
Faculty of Computer Science and Engineering,
Skopje, Macedonia
m.simjanoska@gmail.com, {ana.madevska.bogdanova, marjan.gushev}@finki.ukim.mk

***Abstract*—The exponential growth of biomedical data requires large storage databases and computing resources. Most of the bioinformatics tools as image analysis, data mining, protein folding, and gene sequencing require high computing resources. However, the need for computing capacity varies in different stages of computation. In this paper we present a new Microarray Data Analysis Tool which requires both computing and storage resources. As an appropriate solution we propose a design that can be deployed in a Cloud and offered as SaaS.**

***Index Terms*—DNA, Microarray Analysis, Web Application, Cloud Computing**

I. Introduction

Scientific computing involves the construction of mathematical models and numerical solution techniques to solve scientific and engineering problems that often require a huge number of computing resources to perform large scale experiments or to cut down the computational complexity into a reasonable time frame [1]. The image analysis, data mining, protein folding, and gene sequencing are important tools for biomedical researchers and examples of high compute and resource intensive scientific applications [2]. When comparing the DNA sequencing throughput to the computer speed, sequencing is winning at a rate of about 5-fold per year [3], while computer performance generally follows the Moore's Law, doubling only every 18 or 24 months [4]. The exponential growth of biomedical data requires large storage databases and computing resources. However, the need for computing capacity in the biomedical applications varies dramatically for different stages, i.e. sometimes very big computing power with huge storage space is needed, whereas in the following stage these computationally expensive applications may not require as much computing power as in the previous steps [5]. Considering the fact that the computing resources do not need to be continuously maintained at the maximum capacity, and that DNA sequencing is getting cheaper more quickly than data storage or computation, an appropriate solution for the genome informatics might be to migrate to the cloud [3].

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [6]. In a cloud computing environment, the applications, data and software no longer exist on the client side, instead they are treated as abstract services and reside in the cloud [7]. Cloud's service can be grouped into three categories: software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). SaaS refers to providing on demand applications over the Internet [8].

In this paper we propose a new tool for microarray data analysis developed according to cloud's principles and deployed as SaaS. Considering the permanent growth of microarray experiments and the examination of thousand of genes, an application that can be approached immediately using only a web browser, avoiding the installation of a specific software and complex hardware requirements, would be a great advantage for the scientists.

The rest of the paper is organized as follows. In Section II we present some of the work related to our problem. The whole idea, the tool's design, architecture and implementation are presented in Section III. In the final Section IV we present our conclusions and ideas for future work.

II. Related Work

In this section we present some of the latest work related to our problem.

The idea for this kind of application arises from our previous researches where we explored different kind of microarray technologies. In [9] we developed original statistical methodology for Illumina based experiments, suitable for Bayesian classification. Even though, both Illumina and Affymetrix are widely used microarray technologies, in [10] we showed that both platforms require different statistical approach. Therefore, in [11] we upgraded the methods and proposed new methodology for the Affymetrix based data, also applicable for Bayesian classification analysis. Hereupon, in this paper we organize the developed methodologies for both the Illumina and the Affymetrix platforms, and present an architecture for a novel microarray data analysis tool (MDAT) that uses the advantages of the cloud computing's paradigm.

The authors in [12] motivated by the need to discover cancer-associated eQTLs (expression levels regulators of mRNAs) through integration of two high-dimensional genomic data types (gene expression and genotype), that require more than 13 billion distinct statistical computations, demonstrated that cloud computing is a viable and economical technology that enables large scale data integration and analysis for studies in genomic medicine. In [13] authors present their experience in applying two new Microsoft technologies Dryad and Azure to three bioinformatics applications, and comparison with traditional MPI and Apache Hadoop implementation. Interestingly, authors in [14] present a new desktop application for push-button automated sequence analysis which supports the use of remote cloud computing resources to improve performance for large-scale sequence processing. Another research is presented in [15] where the experiments performed, using the Amazon Elastic Compute Cloud (EC2) infrastructure, have demonstrated that by using distributed implementation of a classification system it is possible to obtain a suitable computing cloud platform for high-dimensional classification problems.

III. Architecture, Design and Implementation

In this section we describe the MDAT's architecture, design and its implementation.

A. *MDAT's Architecture*

When developing an appropriate architecture, we take into account few characteristics necessary for the software to follow the SaaS model. In [16] there is a clear distinction between an ordinary web application and SaaS solution. In order MDAT to be successful SaaS, it must be able to accommodate different users of the application while making it appear to each that they have the application all to themselves, i.e., it must support multi-tenancy. Furthermore, MDAT needs to have the ability to scale up as the number of users grow and need to support a level of basic customization for each tenant. Since MDAT is a scientific tool, it is exposed to various workloads. Handling various amount of input and still maintaining sustainable performance, requires an appropriate design and cloud infrastructure.

Figure 1 presents the users-providers interaction. SaaS provider delivers software services online and allows remote access to the MDAT through the Internet. Thus, MDAT's users are not concerned of any technical details. All the infrastructural resources SaaS providers rent from the IaaS providers according to the pay-per-use pricing model. IaaS providers offer a pool of computing resources that can be dynamically assigned to multiple resource consumers and can easily expand its service to large scales in order to handle rapid increase in service demands [8], and thus preserve the performance.

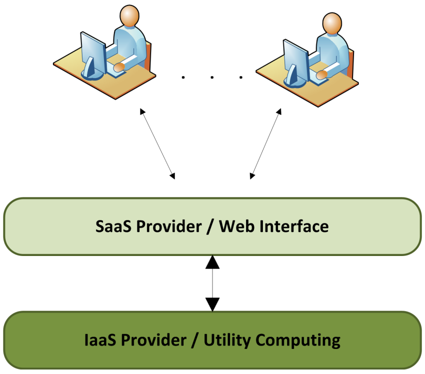
SaaS Provider / Web Interface
IaaS Provider / Utility Computing

Fig. 1. Users and providers

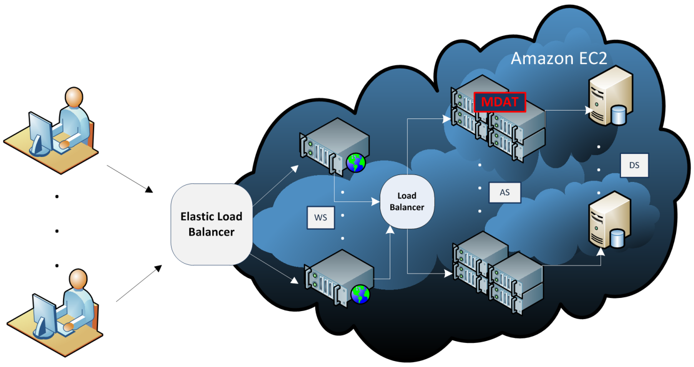

Fig. 2. MDAT Architecture

When presenting the MDAT's architecture in Figure 2, we chose to use Amazon Elastic Computing Cloud (EC2), since it is the largest commercial computing cloud in production among all commercial cloud computing services that can be used for scientific computing. The service is elastic in the sense that it enables the user, which in this case is the software provider, to extend its infrastructure by launching or terminating new virtual machine instances [17]. Once the software provider chooses its configuration, it hosts the MDAT application on an application server, and the users can access it simply using only a web browser. As depicted in Figure 2, IaaS provider takes care of the load by introducing a load balancer, and allows resource's auto-scale if necessary. Therefore, the performance will stay preserved and there will not be any underutilized resources.

B. *MDAT's Design*

Once we defined the MDAT architecture, we proceed to explain its design details.

In Section II we gave a brief review of our previous work related to two widely used platforms for microarray analysis, Illumina and Affymetrix. Both platforms analyse thousands of genes for the purpose of discovering the reasons for some disease occurrence. We used these DNA chips to investigate the genes behaviour when the patients are diagnosed with colorectal cancer. Using data from both Illumina and Affymetrix retrieved from the Gene Expression Omnibus database [18] we realized that both require different statistical approach and for each platform we developed different statistical methodology which can be used for unveiling the genes that show significant expression in presence of colorectal cancer, i.e., the biomarkers. Furthermore, under the assumption that the biomarkers are able to distinguish cancerous from healthy patient, we developed generative model for assigning each biomarker different probability distribution. Once we modelled the prior distributions for both the cancer and the healthy classes, we followed Bayesian posterior probability approach for classifying new patients and obtained very accurate results.

Thus the methodology can be separated in two different stages, the first one is the process for unveiling the biomarkers, and the second is the classification process itself.

The process for uncovering the biomarkers consists of the following steps, mainly the same for both Illumina and Affymetrix platforms:

- *Normalization.* Our aim is to unveil the difference in gene expression levels between the carcinogenic and healthy tissues. We assume that only a small set of genes are differently expressed compared to the biomarker genes, i.e., most of the genes are not correlated to the colorectal cancer. In such cases Quantile normalization (QN) is a suitable normalization method, because it makes the distribution of the gene expressions as similar as possible among each other across all samples [19].
- *Filtering methods.* Since some genes may not be well distributed over their range of expression values, i.e. low expression values can be seen in all samples except one [20]. In order to remove the genes with almost ordered expression levels, we used an entropy filter which measures the amount of information (disorder) about the variable.
- *Paired-sample t-test.* Knowing the facts that both the carcinogenic and healthy tissues are taken from the same patients, and that the whole-genome gene expression follows normal distribution [21], we used a paired-sample t-test. Assuming that the most of the genes do not have different expressions, the null hypothesis states that there is no statistical difference between the carcinogenic and the healthy samples. The rejection of the null hypothesis depends on the significance level which we determine. We considered the genes as statistically significant for a p-value less than 0.01, which means that the chances of wrong rejection of the null hypothesis is less than 1 in 100.
- *False Discovery Rate.* False Discovery Rate (FDR) is a reduction method that usually follows the t-test. FDR solves the problem of false positives, i.e., the genes which are considered statistically significant when in reality there is not any difference in their expression levels. For a threshold of 0.01 we expect 10 genes to be false positive in a set of 1000 positive genes. The significance in terms of FDR is measured as a q-value. It is described as a proportion of significant genes that turn out to be false positives [22].
- *Volcano Plot.* Both the t-test and the FDR method identify different expressions in accordance with statistical significance values, and do not consider biological significance. The biological significance is measured as a fold change [23] which describes how much the expression level changed starting from the initial value. Fold change is measured as ratio between the two expression intensities and does not take into account the variance of the expression levels. In order to display both statistically and biologically significant genes we used the volcano plot visual tool.

The process for preparing the biomarkers for Bayesian classification significantly differs in few steps. Hereupon, we explain the process for each platform distinctively.

Modelling the a priori distribution for Illumina data:

- *Cross-validation method.* In order to choose the patients from which the classifier will learn, we used the cross-validation method. This method avoids over-fitting by not allowing the overlap between the training and the testing set [24].
- *Hypothesis testing.* Once we chose the training set, we used the Kolmogorov-Smirnov test for equality in distribution of the carcinogenic and the healthy tissues. After the tissues rejected the null hypothesis of having the same distribution, we tested each gene distinctively over the Lognormal, Gamma, and Extreme Value probability distribution.

Modelling the a priori distribution for Affymetrix data:

- *Round-up threshold method.* When observing gene expression values, we noticed that a large percentage of the gene expression values are negative. The authors in [25] explain this phenomena within a few processing steps. One way to remove these genes is to transform all gene expression values below some threshold cut-off value to that threshold value [26]. This method is known as Round-up threshold method.
- *Appropriate tissue selection.* Instead of using the cross-validation method for determining the training and the testing set as we did for the Illumina data, for Affymetrix data we choose the training set according to the distribution skewness factor. If the skewness factors are with opposite signs, then these tissues are involved into the training process.
- *Hypothesis testing.* At first, the two sets of tissues are confirmed to be differently distributed using the Kolmogorov-Smirnov test. Hereupon, we performed statistical tests over the Normal, Lognormal, Gamma, and Extreme Value probability distributions. As we have obtained the probabilities from the testing for each gene distinctively, we chose the distribution whose probability is highest and we assign it to the particular gene.

After we modelled the biomarkers' probability distributions, we used them to calculate the Bayesian posterior probability and make accurate diagnostics for the patients' health condition.

MDAT uses the methodology we previously defined. Figure 3 depicts the whole process of microarray data analysis. At first, the user needs to retrieve microarray data from one of the many biological databases that store microarray experiments. Therefore, the user has to upload the data for further analysis and make a choice between the Affymetrix and the Illumina platform. Once the choice is been made, the format of the data has to be verified according to the chosen platform. Hereupon, the user has an option to unveil cancer biomarkers, or, if the biomarkers are already known, the user can proceed directly to the classification process. Unveiling the biomarkers follows the platform independent procedure specified previously in this section. After the biomarkers are discovered, the user can use them for classification, or, can end the procedure immediately. If one decides to continue to the classification process, choosing an appropriate platform is necessary again because of the different classification methodology for each of them. Once the Bayesian classification produces outcomes, a new document is created for the results, and the process finishes.

C. *MDAT's Implementation*

In this section we give an explanation of the MDAT's implementation.

In order to solve computationally and data-intensive problems using multi-core processors, MATLAB developed Parallel Computing Toolbox which consists of parallel for-loops, special array types, and parallelized numerical algorithms. This toolbox allows the developers to parallelize MATLAB applications without MPI programming. Moreover, once the application is programmed on a multi-core desktop computer, without changing the code, it can be run in the cloud using MATLAB Distributed Computing Server as depicted in Figure 4.

However, previously we claimed that our tool can be used without installing any software or hardware. Therefore, in order the end user to run MDAT independently of MATLAB, the application needs to be compiled using the MATLAB compiler runtime (MCR). The MCR is a standalone set of shared libraries that enables the execution of compiled MATLAB applications or components on computers that do not have MATLAB installed. The compiler produces files which are used by the MATLAB Builder NE which encrypts the MATLAB programs and then generates .NET or COM wrappers around them so that they can be accessed just like native .NET and COM components [27].

Once the MDAT is compiled with MCR and connected to the .NET interface using the MATLAB Builder NE, it can be deployed in the cloud, and all users can use it without installing MATLAB on their local machines.

IV. Conclusion and Future Work

Considering our previous researches, we concluded that different microarray platforms require different statistical analysis. Therefore, we developed two different methodologies for analysis and classification of Affymetrix and Illumina based experiments. In this paper we propose a new tool for microarray data analysis which implements the two distinctive procedures. Since the Microarray Data Analysis Tool (MDAT) requires large storage databases and computing resources, we propose an application architecture that can be deployed in a Cloud and offered as SaaS. Therefore, the application can be approached by the user using only a web browser, avoiding

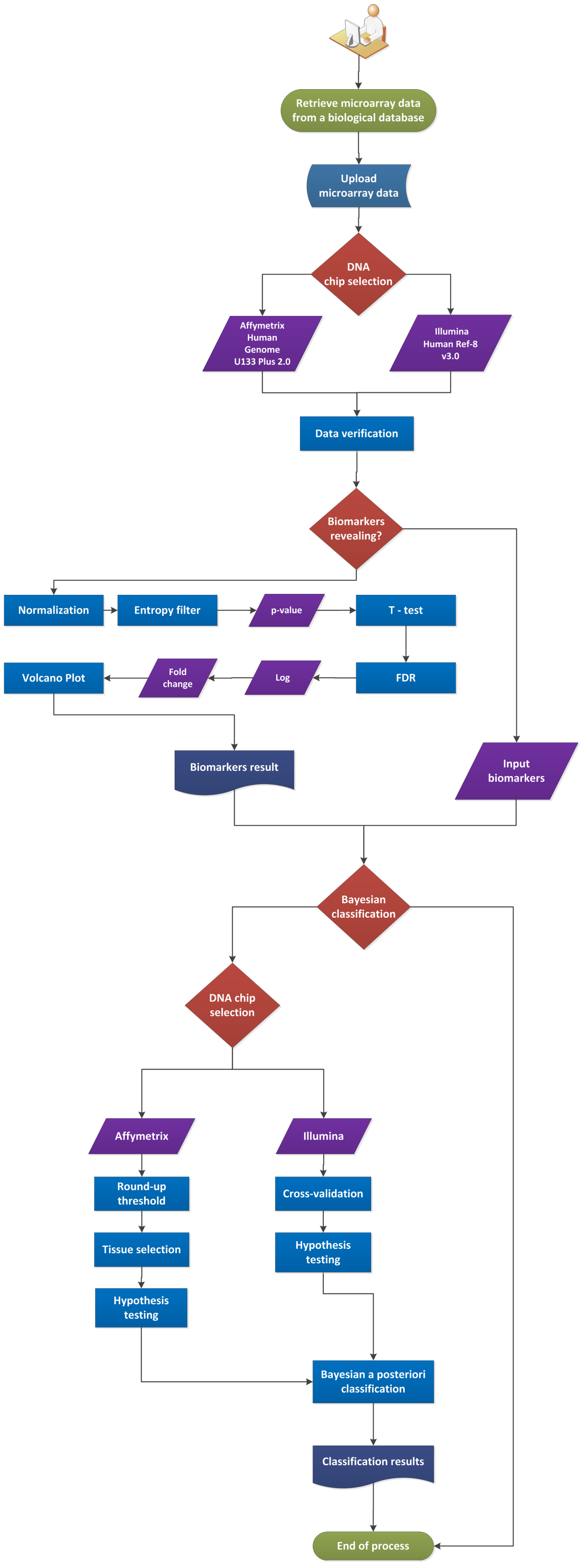

Fig. 3. MDAT Design

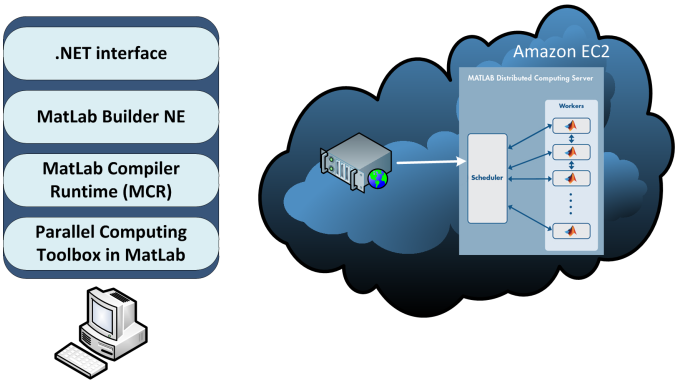
.NET interface
MatLab Builder NE
MatLab Compiler Runtime (MCR)
Parallel Computing Toolbox in MatLab
Amazon EC2
MATLAB Distributed Computing Server
Workers
Scheduler

Fig. 4. MDAT Implementation

the installation of a specific software and complex hardware requirements.

In our future work we will develop the tool we proposed, and we will perform tests if it satisfies the expectations from the cloud implementation.

References

[1] C. Vecchiola, S. Pandey, and R. Buyya, "High-performance cloud computing: A view of scientific applications," in *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*. IEEE, 2009, pp. 4–16.

[2] A. Rosenthal, P. Mork, M. H. Li, J. Stanford, D. Koester, and P. Reynolds, "Cloud computing: A new business paradigm for biomedical information sharing," *Journal of Biomedical Informatics*, vol. 43, no. 2, pp. 342–353, 2010.

[3] L. D. Stein *et al.*, "The case for cloud computing in genome informatics," *Genome Biol*, vol. 11, no. 5, p. 207, 2010.

[4] G. E. Moore *et al.*, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[5] H. Chae, I. Jung, H. Lee, S. Marru, S.-W. Lee, and S. Kim, "Bio and health informatics meets cloud: Biovlab as an example," *Health Information Science and Systems*, vol. 1, no. 1, p. 6, 2013.

[6] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Information Technology Laboratory, Sep. 2011.

[7] L. R. Rewatkar and U. A. Lanjewar, "Implementation of cloud computing on web application," *International Journal of Computer Applications IJCA*, vol. 2, no. 8, pp. 28–32, 2010.

[8] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, vol. 1, no. 1, pp. 7–18, 2010.

[9] M. Simjanoska, A. M. Bogdanova, and Z. Popeska, "Recognition of colorectal carcinogenic tissue with gene expression analysis using bayesian probability," in *ICT Innovations 2012*. Springer, 2013, pp. 305–314.

[10] A. M. Bogdanova, M. Simjanoska, and Z. Popeska, "Classification of colorectal carcinogenic tissue with different dna chip technologies," *the 6th International Conference on Information Technology, ser. ICIT*, 2013.

[11] M. Simjanoska, A. M. Bogdanova, and Z. Popeska, "Bayesian posterior probability classification of colorectal cancer probed with affymetrix microarray technology," *Proceedings of the 36th International Convention, MIPRO, CIS Intelligent Systems*, 2013.

[12] J. T. Dudley, Y. Pouliot, R. Chen, A. A. Morgan, and A. J. Butte, "Translational bioinformatics in the cloud: an affordable alternative," *Genome medicine*, vol. 2, no. 8, p. 51, 2010.

[13] X. Qiu, J. Ekanayake, S. Beason, T. Gunarathne, G. Fox, R. Barga, and D. Gannon, "Cloud technologies for bioinformatics applications," in *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*. ACM, 2009, p. 6.

[14] S. V. Angiuoli, M. Matalka, A. Gussman, K. Galens, M. Vangala, D. R. Riley, C. Arze, J. R. White, O. White, and W. F. Fricke, "Clovr: A virtual machine for automated and portable sequence analysis from the desktop using cloud computing," *BMC bioinformatics*, vol. 12, no. 1, p. 356, 2011.

[15] C. Vecchiola, M. Abedini, M. Kirley, X. Chu, and R. Buyya, "Gene expression classification with a novel coevolutionary based learning classifier system on public clouds," in *e-Science Workshops, 2010 Sixth IEEE International Conference on*. IEEE, 2010, pp. 92–97.

[16] IBM, "Convert your web application to e multitenant saas solution," 2010. [Online]. Available: http://www.ibm.com/developerworks/cloud/library/cl-multitenantsaas/

[17] A. Iosup, S. Ostermann, M. N. Yigitbasi, R. Prodan, T. Fahringer, and D. H. Epema, "Performance analysis of cloud computing services for many-tasks scientific computing," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 6, pp. 931–945, 2011.

[18] "Gene expression omnibus (geo)," 2013. [Online]. Available: http://www.ncbi.nlm.nih.gov/geo/

[19] Z. Wu and M. Aryee, "Subset quantile normalization using negative control features," *Journal of Computational Biology*, vol. 17, no. 10, pp. 1385–1395, 2010.

[20] I. Kohane, A. Butte, and A. Kho, *Microarrays for an integrative genomics*. MIT press, 2002.

[21] Y. Hui, T. Kang, L. Xie, and L. Yuan-Yuan, "Digout: Viewing differential expression genes as outliers," *Journal of Bioinformatics and Computational Biology*, vol. 8, no. supp01, pp. 161–175, 2010.

[22] J. Storey and R. Tibshirani, "Statistical significance for genomewide studies," *Proceedings of the National Academy of Sciences*, vol. 100, no. 16, pp. 9440–9445, 2003.

[23] K. HC and S. AA, "Gene expression profile analysis by dna microarrays: Promise and pitfalls," *JAMA*, vol. 286, no. 18, pp. 2280–2288, 2001.

[24] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of Database Systems*, vol. 5, 2009.

[25] M. Weir and M. Rice, "Microarray clustering analysis." [Online]. Available: https://wesfiles.wesleyan.edu/courses/biol265/microarray_lab.htm

[26] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proceedings of the National Academy of Sciences*, vol. 96, no. 6, pp. 2907–2912, 1999.

[27] MATLAB, 2013. [Online]. Available: http://www.mathworks.com/products/