A Multiagent System for the Analysis of Sequence Data

Roberto González
Computer Science
University of Salamanca
Salamanca (Spain)
rgonzalezramos@usal.es

Javier Bajo
Computer Science
University of Salamanca
Salamanca (Spain)
jbajope@usal.es

Juan F. De Paz
Computer Science
University of Salamanca
Salamanca (Spain)
fcofds@usal.es

Gabriel Villarubia
Computer Science
University of Salamanca
Salamanca (Spain)
gvg@usal.es

Carolina Zato
Computer Science
University of Salamanca
Salamanca (Spain)
carol_zato@usal.es

Juan M. Corchado
Computer Science
University of Salamanca
Salamanca (Spain)
corchado@usal.es

***Abstract*—**The analysis of sequence data requires the processing of the data obtained from sequencers for their subsequent comparison with genomes. The information recovered from the sequencers must be assembled and aligned in order to recover the variations that exist in the patient DNA. This study proposes a system to detect and classify variations by integrating information taken from biomedical databases. The system incorporates different algorithms to search for differences as compared to the reference genome for patients.

Keywords-component; *genetic sequencing, distributed computing, bioinformatics*

I. INTRODUCTION

The process of sequence analysis requires the use of databases to integrate information taken from databases and the information provided by sequencers [11] [12]. Sequencers can obtain sequences from nucleotides. The size of the sequences can vary according to the technology used can vary from a few dozen to hundreds of nucleotides. This increase in information has made it necessary to create systems that can perform a distributed analysis of information and be adapted to different types of analysis such as with genetic sequencing. The information generated by sequencers and the information that already exists in databases require procedures that facilitate the automatic analysis of the data and discard information considered to be irrelevant. To achieve this, it is necessary to create a system than can extract sequences of interest for their posterior analysis, and discard information considered to be irrelevant.

Genetic analysis has changed a great deal in recent decades, having progressed from electronic microscopic analysis to the level of nucleotides. With the appearance of expression arrays, specifically BAC arrays and more importantly Exon arrays [14], it became necessary to create systems that would allow the distributed analysis of information to improve the output of algorithms. The use of NGS (next generation sequencing) has noticeably increased the amount of information, which it is necessary an improvement in the performance and a reduction in execution time of the software. As a result, it has become necessary to create systems that facilitate the management of distributed systems. These systems must facilitate the creation of algorithms that are executed in a distributed way, which enables the dynamic generation of control flows.

This study proposes the use of multiagent systems [18] capable of analyzing information taken from sequencers and integrating it with information from databases The information retrieved from the sequencers must be compared against reference genomes taken from patients that have been previously sequenced. Using localized variations, it is necessary to analyze databases to extract information considered relevant. Within the context of this study, the proposed system focuses on detecting relevant patterns and mutations within the sequence data taken from patient samples provided by the Cancer Institute of the University of Salamanca. The analysis of sequencing data requires various types of processes: i) assembly [13] ii) alignment [13] and iii) knowledge extraction [14] in order to analyze sequence data. The Cancer Institute of the University of Salamanca is

striving to develop tools to automate the evaluation of data and to facilitate the analysis of information. This proposal is a step forward in this direction and the first step toward the development of a multiagent system.

This article is structured as follows: section 2 reviews the state of the art in genetic sequencing; section 3 presents the proposed architecture and adapts the architecture to the case study; section 4 presents the results and conclusions.

II. Massive Analysis and Sequencing

Sequencing began in the 60s, although it was not until the 80s and the Sanger method [13] that gene and genome sequencing emerged. The sequencing process was a laborious manual process; following the development of automated sequencing in the late 80s the volume of information increased dramatically. The process of separating DNA fragments with automated sequencing was initially performed with gel electrophoresis [10], subsequently replaced by capillary electrophoresis [10], after which pyrosequencing [10] was developed. There are currently various types of NGS with different capabilities in base pairs. Zhang et al. [13] describe the different manufacturers. The length of the fragments of the base pairs can vary according to the sequencing used, from 25 bp to the 500 bp used with sequencing by the Roche company, which can perform de-novo sequencing [13]. In the near future, the length of sequenced base pairs is expected to increase considerably; in fact, new research in techniques has developed SMRT (single molecule real time) sequencing, which can achieve 10,000 bp, facilitating the processes of new genome assembly and sequencing.

The human genome is estimated to be about 3,000 million base pairs long and contain around 25,000 genes [19]. Consequently, sequencing genome fragments of 500 bp at a time is costly and requires computational techniques that can join contig fragments to generate the complete genome. Sequencing is not usually applied to just any part of the genome; instead, specific exon sequences corresponding to the DNA code are selected. Exons are the part of the DNA that is represented in the messenger RNA. The regions that are transcribed in the messenger RNA can later be converted into proteins [20], hence the relevance of its analysis and the detection of variants.

The study of variations in the coded regions is of vital interest in determining changes in proteins. Detecting these changes permits an improvement in diagnosis and treatment since proteins regulate the biological behavior of animal [22] and plant [21] organisms. The process of detecting variants requires the application of various algorithms that can compare the sequence data of one patient with a reference genome. The process of analysis is usually carried out by following these steps: assembly, alignment with the reference genome, and analysis of the variations detected. The processes of both assembly and alignment have been widely researched, resulting in the existence of many algorithms [13]. However, there are no semi-automated processes to facilitate the analysis of the detected variations. As a result, the process is performed manually by searching different data bases, making it quite costly with regards to both personnel and time.

The most interesting types of possible variations to analyze this type of problem are as follows:

1. Point mutation: Change of a single nucleotide
    a. SNP (Single Nucleotide Polymorphism): a mutation of a known nucleotide shared by more than 1% of the population.
    b. Mutation: genetic alteration of a nucleotide that does not correspond to a SNP.
2. Large scale mutation:
    a. Deletion: nucleotides within the DNA sequence of the reference genome do not appear in the genome that has been analyzed, and are substituted by gaps as needed to carry out the alignment.
    b. Translocation: a section of the chromosome is inserted into the chromosome being analyzed. A fragment of a different chromosome is kept between the two sequences of chromosomes under study.
    c. Insertion: similar to translocation, but in this case the sequence of the chromosome under study does not appear after inserting a different chromosome.

Certain types of variations such as translocations are complicated to analyze with a sequence analysis if only very specific genes are analyzed; this is because fragments analyzed are not generally long enough to contain translocations.

III. Proposed Architecture

This study proposes a multiagent system [18] that incorporates CBR (Case-based reasoning) agents to retrieve and subsequently classify the variations detected during the sequencing process. The purpose of CBR is to solve new problems by adapting solutions that have been used to solve similar problems in the past [28]. The primary concept when working with CBRs is the concept of case. A case can be defined as a past experience, and is composed of three elements: a problem description which describes the initial problem, a solution which provides the sequence of actions carried out in order to solve the problem, and the final state which describes the state achieved once the solution was applied. A CBR manages cases (past experiences) to solve new problems. The way cases are managed is known as the CBR cycle, and consists of four sequential steps which are recalled every time a problem needs to be solved: retrieve, reuse, revise and retain. Each of the steps of the CBR life cycle requires a model or method in order to perform its mission. The algorithms selected for the retrieval of cases should be able to search the base case and select the problem and corresponding solution most similar to the new situation. Once the most important cases have been retrieved, the reuse phase begins, in which the solutions for the retrieved cases are adapted and a new solution is generated. The revise phase consists of an expert revision

for the proposed solution. Finally, the retain phase allows the system to learn from the experiences obtained in the three previous phases, consequently updating the cases memory.

The architecture created for this study contains three separate layers: the administration layer contains agents assigned to maintain the algorithms specific to the case study; the control layer contains agents responsible for controlling the system; and the specialization layer contains the agents and the processes specific to the case study.

Figure 1 displays the agents that correspond to the coordination and control layer. These layers are independent of the case study, allowing their functionality to be reused.

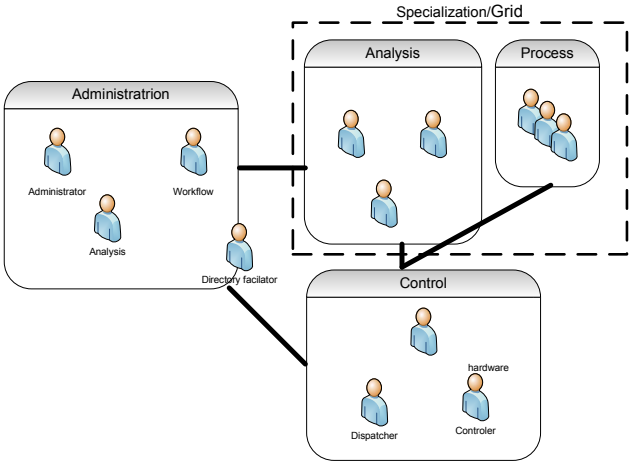

Figure 1. Architecture for system coordination and control

. The coordination layer includes the administrator, analysis, workflow and directory facilitator agents. The administrator agent is in charge of storing and controlling project data for their subsequent analysis. Each project contains information about the flow of analysis and the results obtained by the applied algorithms. Additionally administrator agent includes all associated roles including the status of the project process, launched tasks, and the type of data associated with the different case studies. The analysis agent is in charge of recovering previous flows of analysis and storing new flows of analysis that may be used to recommend flows of execution with similar data. The workflow agent is responsible for creating new flows of execution based on existing algorithms for different types of analyses. Finally, the directory facilitator (DF) stores and administers existing algorithms for each of the possible types of analysis; its functionality is similar to that of a web services DF.

The control layer includes the agents responsible for controlling the state of execution for the GRID. The dispatcher, hardware and controller agents are available in this layer. The dispatcher agent is in charge of recovering and distributing tasks between the nodes. The hardware agent controls machine resources available on the GRID. The controller agent controls the machine load level to control the state of execution for the machines containing GRID.

The specialization layer is composed of agents and processes that are executed in the GRID nodes. These processes and agents are specific to each case study and are responsible for defining the hardware needs for their execution, and breaking the tasks down into subtasks that are sent to the dispatcher.

A. Sequence Analysis

The process of sequence data analysis varies according to the results that one wants to obtain. It normally requires a process of assembly, alignment and knowledge extraction to automatically process the data. As the architecture must be specific to this end, agents and processes specialized in performing these tasks are required. The agents are responsible for establishing the restrictions and procedures for distributing tasks along the GRID nodes according to available resources.

The specialization layer in this case study was composed of the following agents: assembly, alignment and knowledge extraction. Each agent defines the following roles for the purpose of carrying out the task for which they were added to the system: manage available algorithms to execute the task; manage the resources needed to apply each algorithm; determine the preconditions for executing tasks; manage the nodes required to execute the tasks; break the tasks down into subtasks that are subsequently queued in the dispatcher.

Each agent in this stage has various processes that are executed through GRID in a distributed manner. The processes can vary according to the algorithm that is selected in the work flow of the project created in the analysis. Thus, the algorithms developed for each stage of the case study are as follows:

1) Assembly

The assembly process varies according to the size of each reading that is used. In this particular case, we chose to use the algorithm provided by the manufacturer of the sequencer being used. Different assembly algorithms can be seen in [13] and [27]. Roche provides the Newbler assembler [26], which was used in this study.

2) Alignment

The alignment process consists of establishing the fragment of the reference genome that is most similar to the fragment of the patient being treated. The alignment algorithms are applied to different fields in addition to bioinformatics.

While there are many different ways to carry out the alignment process, performance is ultimately the most important factor. The alignment algorithms used are local, since the sequence to be aligned, or the contigs, is smaller in size than the reference genome. Local alignments are based on the Smith-Waterman algorithm [23]. The alignments can be given in pairs or groups according to the number of fragments that must be analyzed simultaneously. There are currently many alignment algorithms, but the most commonly used are BLAST (Basic Local Alignment Search Tool) [24] and BLAT (BLAST-Like Alignment Tool) [25].

BLAT can perform an alignment faster than BLAST, but it cannot ensure that the final alignment is the best one possible, although performance is greatly improved. Additionally, there are many algorithms that can be found in different review articles such as [13] and [27].

3) Extraction of knowledge

The classification algorithms can be divided in: decision trees, decision rules, probabilistic models, fuzzy models, based on functions, ensemble. During the extraction of knowledge phase, different analyses of the variations were performed to detect the following types of alterations: SNP, point mutation not SNP, insertions and deletions. The process of detecting SNP and other point mutations is simple since it only involves searching information in databases that contain the previously published information. The problem lies in the detection of the insertions, given that the human genome contains homologies, and DNA strands are repeated in different regions. In order to detect the insertions and deletions, different classification techniques to facilitate automatic detection were applied. The classification algorithms included decision trees, decision rules, probabilistic models, fuzzy models, function-based algorithms and ensemble. The system selects these algorithms for each kind of method: decision rules RIPPER [4], One-R [9], M5 [7], decision trees J48 [8], CART [2] (Classification and Regression Trees), probabilistic models naive Bayes [5], fuzzy models K-NN (K-Nearest Neighbors) [1], Support Vector Machine (SVM) [29] and finally ensemble such as Bagging [3] and Ada-Boosting [6].

The Support Vector Machine (SVM) is a supervised learning technique applied to the classification and regression of elements. SVM can be applied in a variety of fields such as chemistry, ambient intelligence, modelling and simulation, and data or text mining. The algorithm represents an extension of the linear models [29]. Originally developed for the classification of linearly separable problems, it basically consists of finding the straight line or hyper plane (in two or more dimensions) that makes it possible to separate the elements of a set. SVM can also separate different classes of elements that cannot be separated linearly. To do so, it uses functions to map out the initial space of coordinates in a highly dimensional space. Because the dimensionality of the new space can be so high, it is not practical to calculate the hyperplanes that perform the linear separation. Instead, a series of non linear functions known as kernel Φ are used.

The following equation is used to perform the classification (1) [30].

$$class(x_k) = signe\,[w\Phi(x_k) + b] = signe\left(\sum_{i=1}^{m} \lambda_i y_i \Phi(x_i)\Phi(x_k) + b\right) \quad (1)$$

Where x_i is a vector with n-dimension, the idea is to convert the elements x_i in a highly dimensional space using the application of a feature function $\Phi(x)$, λ_i is a Lagrange multiplier, and y_i is the output value for the pattern b constant. The calculation of these values is described in [31]

As we can see, there is a product $\Phi(x_i)\Phi(x_k)$ that, according to the dimensionality of the new space, can be very costly to calculate. For this reason, it is necessary to select a series of kernel functions that can operate in the original space to perform these calculations without requiring a heavy computational load.

To calculate the classifier $class\,(x_k)$ there are algorithms such as the Sequential Minimal Optimization (SMO) [32]. From the hyperplane calculated by SMO, we proceed to calculate the distance of each of the points to the hyperplane. These distances will be calculated to estimate the error in the calculation of the distance and to make the mixture of methods as described in the last paragraph of the subsection classification model. The distance is calculated according to equation (2)

$$d(x; w, b) = \frac{|\, w \cdot \Phi(x) + b \,|}{\|w\|} \quad (2)$$

IV. Results and Conclusions

Genetic sequencing was applied to a data set taken from patients with leukemia. Specific genes were sequenced from a total of 8 patients, each of whom had approximately 110,000 sequence fragments that corresponded to the regions relevant to this study. The sequenced fragments vary in length for the different patients, as shown in figure 2.

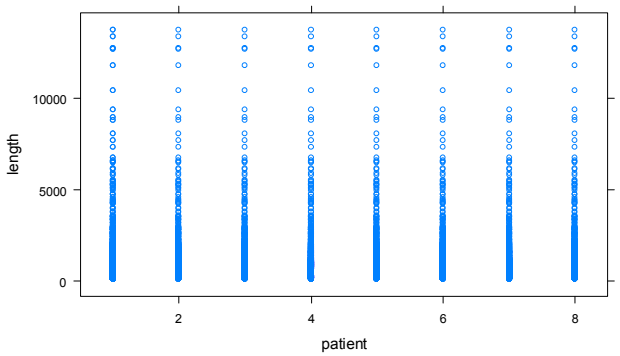

Figure 2. Box diagram of the lengths of the fragmented segments.

The version of the reference genome used in this study corresponds to HG18; this is because the information used as a reference in selecting the sequence regions was obtained in previous studies using the same version.

The system was tested to validate the generic architecture proposed and to validate the system's capacity for analyzing the proposed case study. The main goal in validating the architecture was to determine the efficiency of the system and its ability to distribute tasks according to the existing nodes. To validate the increase in performance, we selected the Newbler, BLAT and SVM workflows. The average sizes for each of the aligned patients are as follows: 1023.134 1356.855 1292.022 1251.492 1239.192 1306.707 1302.321 1362.151.

After completing the assembly process, the fragments were aligned using the BLAT algorithm, which obtained PSL output files. The output format of PSL files was used for the training and prediction of the insertions in the contigs that were analyzed. The specific files used to predict the alterations are the following: matches, misMatches, repMatches, nCount, qNumInsert, qBaseInsert, tNumInsert, tBaseInsert, qSize, qStart, qEnd, tSize, tStart, tEnd, blockCount, blockSizes. To make the prediction, the inputs of the two alignments with the highest number of matches were added.

The output data were analyzed for patient 5, while the outputs were classified manually. The classification was done according to SVM and other techniques. During the classification, an attempt was made to analyze the existence of large scale mutations such as chromosomal translocation. The SVM classifier was then applied to the output of patient 1, and the results obtained are shown in Table 1. Table 1 shows the number of elements classified for each. To evaluate the significance of the possible classification techniques used during the reuse phase, we performed a comparison between different classifiers.

TABLE I. CLASSIFICATION PROCESS

	True positive	False positive	False negative	True negative
SVM	7	0	0	1565
J48	7	1	0	1564
JRip	6	2	1	1563
NaiveBayes	7	1	0	1564

V. CONCLUSIONS

SVM improves the results provided by the other methods; however, the differences are not very significant since the majority of cases correspond to the True negative category.

The final number of alterations in the patient are 6584 SNPs and 2021 Unknown variants. The number of unknown alterations (insertions and deletions) is high due to the pathology and the analyzed regions. These alterations are analyzed by direct comparison against information taken from databases at UCSC (University of California, Santa Cruz).

The multiagent system has made it possible to integrate algorithms that can adapt to a specific case study, facilitating the distributed execution of work flows. The system facilitates the integration of algorithms for different case studies and reduces the execution time in an efficient manner, so long as it remains possible to improve performance by separating tasks for their more effective execution in GRID technology.

ACKNOWLEDGMENT

This work was supported by the Spanish Ministry of Science TIN 2009-13839-C03-03 Project..

REFERENCES

[1] Aha D., Kibler D., Albert, M.K.: Instance-based learning algorithms. Machine Learning. vol. 6, 37-66 (1991).

[2] Breiman, L., Fried, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth International Group. (1984).

[3] Breiman, L.: Bagging predictors. Machine Learning. vol. 24 (2), 123-140 (1984).

[4] Cohen, W.W.: Fast effective rule induction. In Proceedings of the 12th International Conference on Machine Learning. Morgan Kaufmann. 115-123 (1995).

[5] Duda R.O., Hart P. Pattern classification and Scene Analysis. New York: John Wisley & Sons. (1973).

[6] Freund, Y., Schapire, R.E. Experiments with a new boosting algorithm. In Thirteenth International Conference on Machine Learning. 148-156 (1996).

[7] Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning. vol. 11, 63-91 (1993).

[8] Quinlan, J.R.: C4.5: Programs For Machine Learning. Morgan Kaufmann Publishers Inc. (1993)

[9] Holmes, G., Hall, M., Prank, E.: Generating Rule Sets from Model Trees. Advanced Topics in Artificial Intelligence. vol. 1747/1999, 1-12 (2007).

[10] Keith R. Mitchelson, David B. Hawkes, Rustam Turakulov and Artem E. Men Chapter Overview: Developments in DNA Sequencing, Perspectives in Bioanalysis, Volume 2, 2007, Pages 3-44

[11] Ross A. Soo, Ling Zhi Wang, Swee Siang Ng, Pei Yi Chong, Wei Peng Yong, Soo Chin Lee, Jian Jun Liu, Tai Bee Choo, Lai San Tham, How Sung Lee, Boon Cher Goh, and Richie Soong, f, Distribution of gemcitabine pathway genotypes in ethnic Asians and their association with outcome in non-small cell lung cancer patients, Lung Cancer, Volume 63, Issue 1, January 2009, Pages 121-127

[12] F. Esteban, J.L. Royo, M.A. González-Moles, A. Gonzalez-Perez, M. Redondo, R. Moreno-Luna, M. Rodríguez-Sola, A. Gonzalez, L.M. Real, A. Ruiz and R. Ramírez-Lorca, CAPN10 alleles modify laryngeal cancer risk in the Spanish population, European Journal of Surgical Oncology (EJSO). Volume 34, Issue 1, January 2008, Pages 94-99

[13] Jun Zhang, Rod Chiodini, Ahmed Badr and Genfa Zhang The impact of next-generation sequencing on genomics, Journal of Genetics and Genomics, Volume 38, Issue 3, 20 March 2011, Pages 95-109

[14] Juan M. Corchado, Juan F. De Paz, , Sara Rodríguez, and Javier Bajo, Model of experts for decision support in the diagnosis of leukemia patients, Artificial Intelligence in Medicine, Volume 46, Issue 3, July 2009, Pages 179-200

[15] F. Gregoretti, G. Laccetti, A. Murlib, G. Olivaa, and U. Scafuri MGF: A grid-enabled MPI library, Future Generation Computer Systems, Volume 24, Issue 2, February 2008, Pages 158-165

[16] Haoqiang Jin, Dennis Jespersen, Piyush Mehrotra, Rupak Biswas, Lei Huang and Barbara Chapman, High performance computing using

MPI and OpenMP on multi-core parallel systems, Parallel Computing. In Presss

[17] P.S. Rakić, D.D. Milašinović, Ž. Živanov, Z. Suvajdžin, M. Nikolić and M. Hajduković, MPI–CUDA parallelization of a finite-strip program for geometric nonlinear analysis: A hybrid approach, Advances in Engineering Software, Volume 42, Issue 5, May 2011, Pages 273-285

[18] M. Wooldridge, Introduction to MultiAgent Systems, John Wiley & Sons, 2002.

[19] International Human Genome Sequencing Consortium, Finishing the euchromatic sequence of the human genome, Nature 431, 931-945 (21 October 2004)

[20] Sitanshu Sekhar Sahua, and Ganapati Pand, Identification of Protein-Coding Regions DNA Sequences Using A Time-Frequency Filtering Approach. Genomics, Proteomics & Bioinformatics. Volume 9, Issues 1-2, April 2011, Pages 45-55

[21] Bryan Delaney, James D. Astwood, Helen Cunny, Robin Eichen Conn, Corinne Herouet-Guicheney, Susan MacIntosh, Linda S. Meyer, Laura Privalle, Yong Gao, Joel Mattsson, Marci Levine, and ILSI International Food Biotechnology Committee Task Force on Protein Safety, Evaluation of protein safety in the context of agricultural biotechnology, Food and Chemical Toxicology, Volume 46, Supplement 2, May 2008, Pages S71-S97

[22] Scott W. Blume, Nateka L. Jackson, Andra R. Frost, William E. Grizzle, Oleg D. Shcherbakov, Hyoungsoo Choi and Zheng Meng Northwestern profiling of potential translation-regulatory proteins in human breast epithelial cells and malignant breast tissues: evidence for pathological activation of the IGF1R IRES, Experimental and Molecular Pathology, Volume 88, Issue 3, June 2010, Pages 341-352

[23] Ali Khajeh-Saeed, Stephen Poole and J. Blair Perot Acceleration of the Smith–Waterman algorithm using single and multiple graphics processors, Journal of Computational Physics, Volume 229, Issue 11, 1 June 2010, Pages 4247-4258

[24] Stephen F. Altschul1, Warren Gish, Webb Miller, Eugene W. Myers and David J. Lipman, Basic local alignment search tool, Journal of Molecular Biology, Volume 215, Issue 3, 5 October 1990, Pages 403-410

[25] W.J. Kent, BLAT—the BLAST-like alignment tool, Genome Res. 12 (2002), pp. 656–664.

[26] M. Margulies, M. Egholm, W.E. Altman, S. Attiya, J.S. Bader, L.A. Bemben, J. Berka, M.S. Braverman, Y.J. Chen, Z. Chen, S.B. Dewell, L. Du, J.M. Fierro, X.V. Gomes, B.C. Godwin, W. He, S. Helgesen, C.H. Ho, G.P. Irzyk, S.C. Jando, M.L. Alenquer, T.P. Jarvie, K.B. Jirage, J.B. Kim, J.R. Knight, J.R. Lanza, J.H. Leamon, S.M. Lefkowitz, M. Lei, J. Li, K.L. Lohman, H. Lu, V.B. Makhijani, K.E. McDade, M.P. McKenna, E.W. Myers, E. Nickerson, J.R. Nobile, R. Plant, B.P. Puc, M.T. Ronan, G.T. Roth, G.J. Sarkis, J.F. Simons, J.W. Simpson, M. Srinivasan, K.R. Tartaro, A. Tomasz, K.A. Vogt, G.A. Volkmer, S.H. Wang, Y. Wang, M.P. Weiner, P. Yu, R.F. Begley and J.M. Rothberg, Genome sequencing in microfabricated high-density picolitre reactors, Nature 437 (2005), pp. 376–380

[27] Jason R. Miller, Sergey Koren and Granger Sutton, Assembly algorithms for next-generation sequencing data, Genomics, Volume 95, Issue 6, June 2010, Pages 315-327

[28] J. Kolodner, Case-Based Reasoning, Morgan Kaufmann 1993.

[29] Vapnik, V. Lerner, A. Pattern Recognition Using Generalized Portrait Method, in, 1963, pp. 774–780.

[30] Ivanciuc, O. Applications of Support Vector Machines in Chemistry, in: K.B. Lipkowitz, T.R. Cundari (Eds.), John Wiley & Sons, 2007, pp. 291-400.

[31] Garvey, A.J. Lesser, V.R. Design-to-time real-time scheduling, Systems, Man and Cybernetics, IEEE Transactions on, 23 (1993) 1491-1502.

[32] Platt, J.C. Fast training of support vector machines using sequential minimal optimization, (1999) 185-208.