Automatic knowledge extraction in sequencing analysis with multiagent system and grid computing

Roberto González[1], Carolina Zato[1], Rocío Benito[2], Javier Bajo[1], Jesús M. Hernández[2,3], Juan F. De Paz[1,*], Vicente Vera[4], Juan M. Corchado[1]

[1]Department of Computer Science, University of Salamanca Plaza de la Merced, s/n, 37008, Salamanca, Spain

[2]IBMCC, Cancer Research Center, University of Salamanca-CSIC, Spain

[3]Servicio de Hematología, Hospital Universitario de Salamanca, Spain

[4]Department of Estomatology II, Complutense University of Madrid, Plaza Ramón y Cajal, s/n 28040, Madrid, Spain

Summary

Advances in bioinformatics have contributed towards a significant increase in available information. Information analysis requires the use of distributed computing systems to best engage the process of data analysis. This study proposes a multiagent system that incorporates grid technology to facilitate distributed data analysis by dynamically incorporating the roles associated to each specific case study. The system was applied to genetic sequencing data to extract relevant information about insertions, deletions or polymorphisms.

1 Introduction

Advances in genetic sequencing have made it possible to carry out massive sequencing and to perform studies on genetic sequencing in multiple patients [2] [3]. One area of medicine in its height of development and fundamental in the application of techniques that facilitate the automatic treatment of data and the extraction of knowledge is genomics. 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 development of distributed applications and parallel computing currently requires the use of complex software and libraries [7], while some, such as MPI and CUDA [8], even use combinations of libraries. It has become necessary to develop systems that facilitate the creation of distributed systems that allow the distributed implementation and execution of different types of analyses+ by applying technologies such as grid computing [6].

The rise in bioinformatics, primarily following the emergence of microarrays and sequencing in particular, has made it possible to generate great volumes of information [5]. With the appearance of expression arrays, specifically BAC arrays and more importantly Exon arrays [5], it became necessary to create systems that would allow the distributed analysis of information to

*To whom correspondence should be addressed. Email: fcofds@usal.es

improve the output of algorithms. The use of NGS (next generation sequencing) has noticeably increased the amount of information, which has in turn led to an improvement in output and a reduction in execution time. 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 [9] capable of distributed execution performed by the integration of grid technology [6]. The multiagent system integrates agents to manage the roles in the case study that is being carried out. Within the context of this study, the proposed system focuses on detecting relevant patterns and mutations, insertions, deletions or polymorphisms, 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 [4] ii) alignment [4] and iii) knowledge extraction [5] 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 state of the art of multiagent systems and virtual organizations section 4 presents the proposed architecture and adapts the architecture to the case study; section 5 presents the results and conclusions.

2 Massive Analysis and Sequencing

Sequencing began in the 60s, although it was not until the 80s and the Sanger method [4] 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 [1], subsequently replaced by capillary electrophoresis [1], after which pyrosequencing [1] was developed. There are currently various types of NGS with different capabilities in base pairs. Zhang et al. [4] 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 denovo sequencing [4]. 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 [10]. 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 [11], 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 [18] and plant [17] 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 [4]. 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.

3 Virtual Organizations and Multi-Agent Architectures

Nowadays, having software solutions at one's disposal that enforce autonomy, robustness, flexibility and adaptability of the system to develop is completely necessary. The dynamic agents organizations that auto-adjust themselves to obtain advantages from their environment seem a more than suitable technology to cope with the development of this type of system. These organizations could appear in emergent or dynamic agent societies, such as grid domains, peer-to-peer networks or other contexts where agents dynamically group together to offer compound services. However, although on a theoretical level, technology seems to be able to allow the development of this new kind of system, it is still necessary to investigate theories, models, mechanisms, methods and tools in order to develop systems with reorganization capabilities that provide them with the ability to adapt themselves to environmental changes.

Multi-agent systems are characterized by being autonomous, reactive, proactive, socially skilled, etc., [19] and therefore are perfectly suited for creating organizational models in which each agent can send and receive messages with an individual, organization or actor in the real society that is being modelled [20]. Additionally, the interaction between agents can correspond to interactions existing in the real world [20]. Once the model has been established, a simulation is initiated and the system's behaviour is observed. If the agents possess adequate learning and adaptation capabilities, it is possible to obtain knowledge and detect behavioural patterns. There are many agent-based social simulation models that try to analyze different social phenomena [22] [20]. Schelling [21] carried out the first agent-based social simulation in which each person was represented by an agent and the interaction between these agents represented relevant social processes. Epstein and Axtell [22] developed the first large-scale agent-based model, Sugarscape, which allows entire societies to be modelled and different social processes such as death, sickness, sex, reproduction, culture, conflict, war, etc., to be observed. Nevertheless, there is still much work to be done, especially in the field of automated re-organization of virtual organizations [20]. In the framework of this research, we are especially interested in organizations characterized by the need of classification of massive amounts of data and the integration of multi-agent systems and web services can be highly recommended for this.

The integration of multi-agent systems with SOA and Web Service approaches has recently been explored [3]. Some developments are centred on communication between these models, while others are centred on the integration of distributed services, especially Web Services, into the structure of the agents. Oliva et al. [23] have developed a Java-based framework to create

SOA and Web Services compliant applications, which are modelled as agents. Communication between agents and services is performed by using what they call ”artifacts” and WSDL (Web Service Definition Language). Shafiq et al. [24] propose a gateway that allows interoperability between Web Services and multi-agent systems. Liu et al. [25] propose a multi-agent architecture to develop inter-enterprise cooperation systems using SOA and Web Service components and communication protocols. Walton et al. [27] present a technique to build multi-agent systems using Web Services, defining a language to represent the dialogs among agents. There are also frameworks, such as Sun's Jini and IBM's WebSphere, which provide several tools to develop SOA-based systems. Jini uses Java technology to develop distributed and adaptive systems over dynamic environments. Rigole et al. [26] have used Jini to create agents on demand in a home automation system, where each agent is defined as a service in the network. WebSphere provides tools for several operating systems and programming languages. However, the systems developed using these frameworks are not open at all because the framework is closed and services and applications must be programmed using a specific programming language that supports their respective proprietary APIs (Application Programming Interface).

These alternatives have been explored as possible solutions for virtual organizations based on data analysis. We can find several works in bioinformatics that mix the multi-agent system for example in data analysis [28] [32], genomic analysis [29] or in the genomic annotation [30] [31]. Gird computing is commonly used in the bioinformatics, we can see several studies about grid computing in the state of the art of [33]. The use of grid computing and multiagent system is recently studied to develop the load balancing [34] but the objective of the work is to create a multi-agent system to manage the workflow execute in the grid.

These alternatives have been explored as possible solutions for virtual organizations based on data analysis. It is possible to find several works in bioinformatics that use multi-agent systems. For example in data analysis [28] [32], genomic analysis [29] or in the genomic annotation [30] [31]. Grid computing is also commonly used in the bioinformatics. It is possible to find several studies about grid computing in the state of the art of bioinformatics [33]. The use of grid computing and multigent systems has been recently proposed to perform load balancing [34], but the objective of this work is diferent: to create a multi-agent system to manage the workflow execute in the grid. In this sense, the work presented in this paper can be considered as innovative.

4 Proposed SMASasGC Architecture

Bioinformatic data analysis requires different processes and algorithms that vary according to the data recovered. Nevertheless, these different types of analyses all share a common characteristic, namely the high computing cost involved. It has become necessary to develop systems that facilitate the efficient development of distributed systems. In order to analyze distributed data in an efficient manner, grid technology [6] and multiagent systems [9] are integrated to produce a high performance hybrid architecture. The architecture created for this study contains three separate layers: the coordination layer contains agents assigned to maintain the algorithms specific to the case study; the control layer contains agents responsible for controlling the grid; 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.

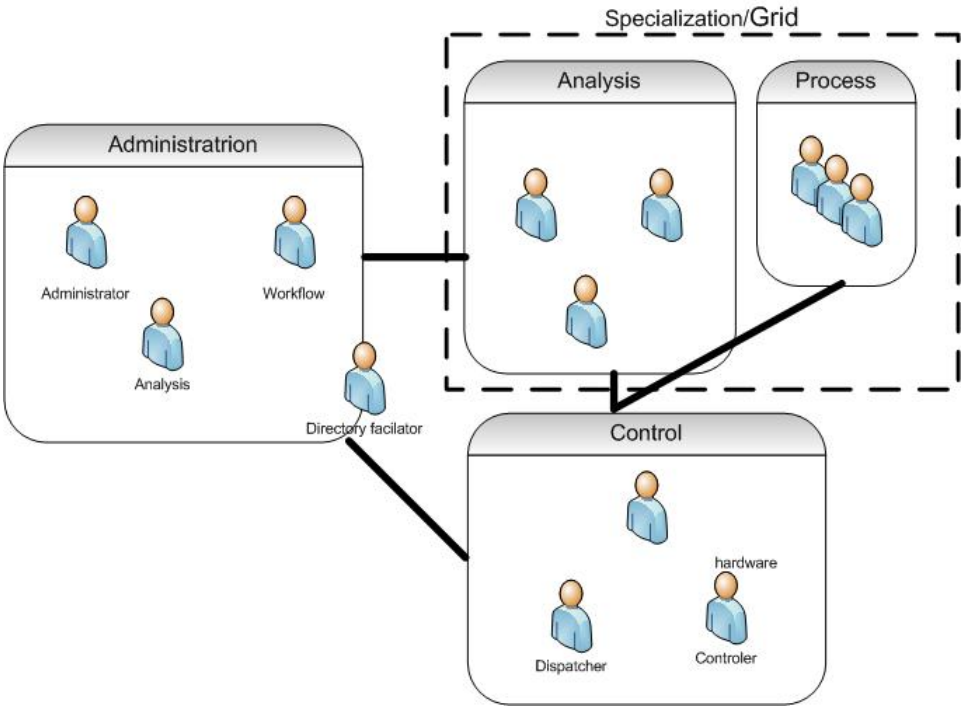

Figure 1: Architecture for system coordination and control layer

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.

The different system agents are distributed according to the layers and the connections that each type of agent can make with the other types of system agents and services. For example, in order to carry out its task, the workflow agent at the administration layer uses a specific sequence to select agents from the analysis layer. In turn, the analysis layer agents select the process that are necessary to carry out the data study: the filtering agent at the analysis layer

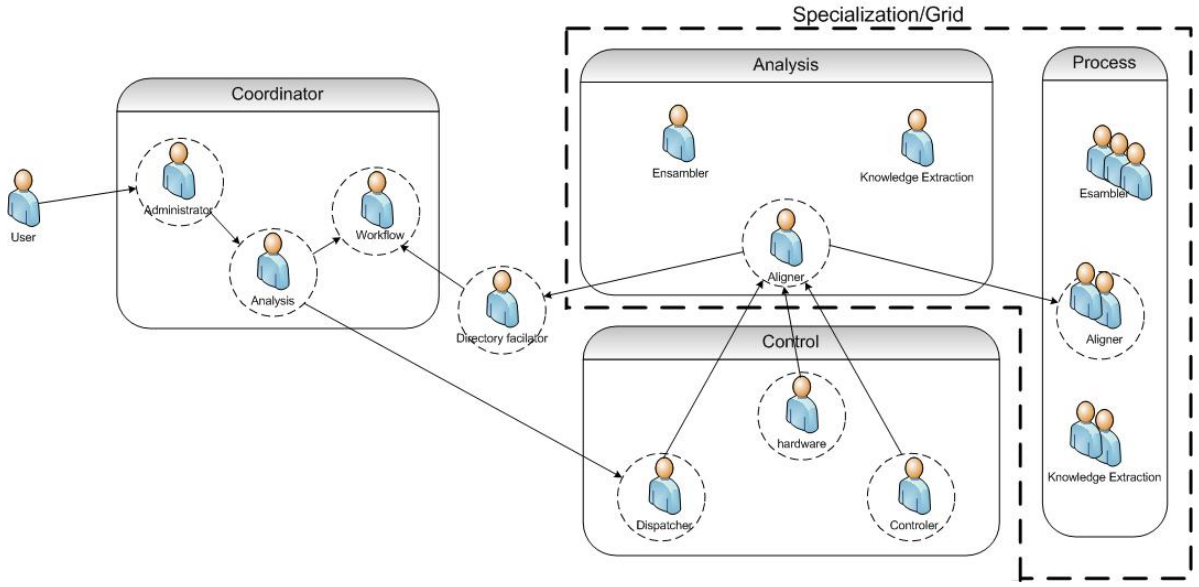

Figure 2: Basic functionality of the system

selects from the services and workflow that are appropriate for the data.

The figure 2 shows the functionality of the system. Basically, the administrator agent creates a project and stores the flow to analyze the data, the agents of the specialization layer break the tasks into subtasks that are sent to the dispatcher. Dispatcher agent distributes the task among the node and the nodes execute the tasks.

4.1 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 workflow of the project created in the analysis.

4.1.1 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 [12]. 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) [13] and BLAT (BLAST Like Alignment Tool) [14]. 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 [4] and [16].

4.1.2 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 [4] and [16]. Roche provides the Newbler assembler [15], which was used in this study.

4.1.3 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 and 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 process of detecting insertions and deletions has a higher level of difficulty because it is necessary to train the classifiers to detect these kind of mutations. The first step consist on classifying several alignment contigs (deletion, insertion or normal) and use the obtained information to train the classifier. After that, the system will use the classifier to classify new contigs. The variables used during the process are described in the results section.

5 Results and conclusions

SAMasGC is web-based system that integrates multi-agent system and grid computing to perform the sequencing analysis. The system incorporates GridGain to carry out the grid computing. The system allows to create a project and to incorporate files to the project. The user can select concretes workflow of analysis of the data, and the system divides the tasks into subtasks in order to distribute the analysis in different nodes using grid computing. The figure 3 shows

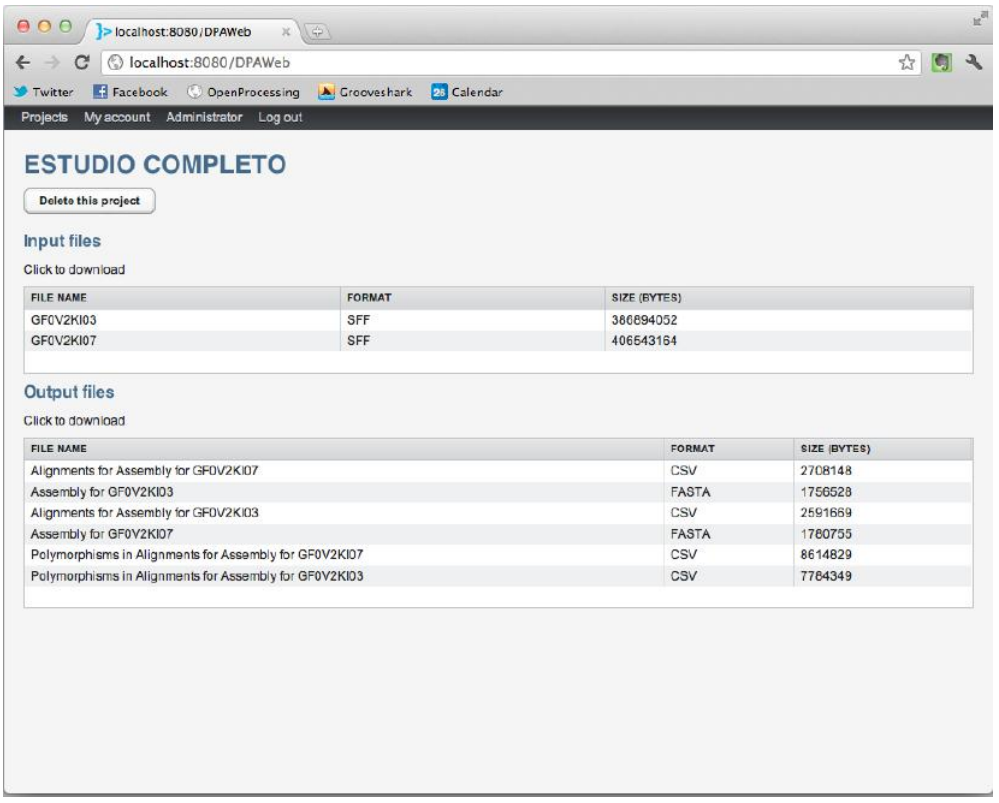

Figure 3: Screenshot of the graphic user interface of SAMasGC

a screenshot of the graphic user interface of the system. The system was tested with 5 core 2 duo P9700 with 4GB of Memory, the computers were connected to an intranet with a transfer rate of 100 Mbit.

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 4.

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. Figure 5 demonstrates the size of the contigs that were assembled once the Newbler algorithm was applied and executed on the grid.

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 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 and blockSizes. The analyzed alterations are insertions, deletions and poly-

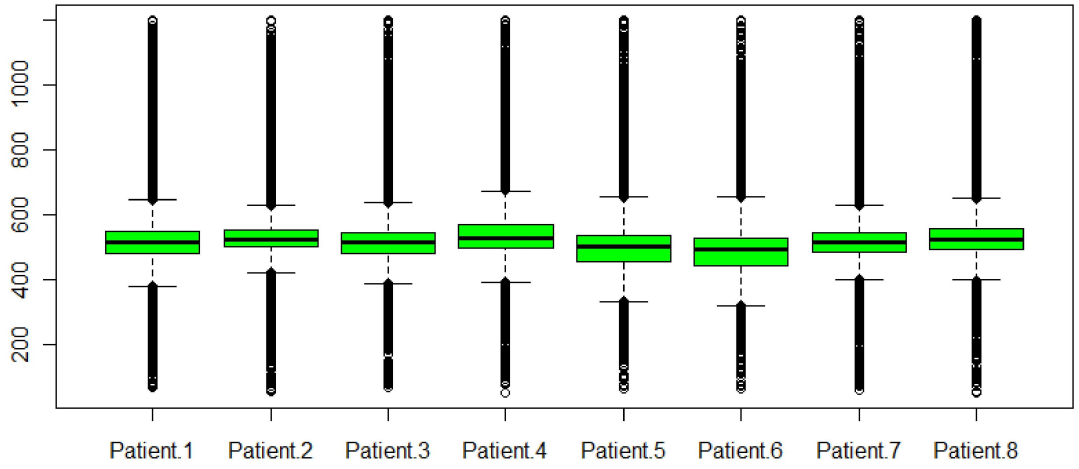

Figure 4: Box diagram of the lengths of the fragmented segments.

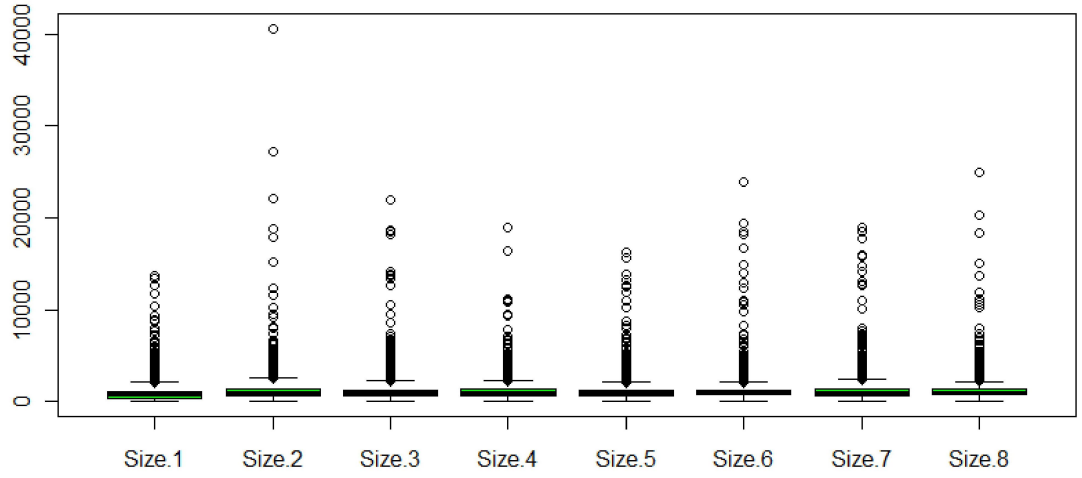

Figure 5: Box diagram with the length of the assembled contigs

morphisms, in order to detect these alterations the system used the SNP130 table from UCSC. The final number of alterations in the patient 1 are shown in table 1, the number of unknown alterations (insertions and deletions) is high due to the pathology and the analyzed regions.

Table 1: Detected variants during the knowledge extraction

Initial	Contigs	Contigs with variants SNPs	Unknown variants
1310	1100	6584	2021

The next step was to analyze the execution time in order to observe the system's scalability according to the number of nodes on which the processes were dropped. The execution time decrease linearly with the number of nodes in the system. The table 2 shows the execution times in seconds for several nodes.

The multiagent system has made it possible to integrate algorithms that can adapt to a specific case study, facilitating the distributed execution of workflows. The system facilitates the integration of algorithms for different case studies and reduces the execution time in an efficient

Table 2: Variants

	1 Node	2 Nodes	3 Nodes	5 Nodes
Segmentation	691s	405s	241s	149s
Polymorphism	2340s	1145s	710s	468s

manner, so long as it remains possible to improve performance by separating tasks for their more effective execution in grid technology.

Acknowledgements

This work has been supported by the MICINN TIN 2009-13839-C03-03.

References

[1] K. R. Mitchelson, D. B. Hawkes, R. Turakulov and A. E. Men Chapter. Overview: Developments in DNA Sequencing. *Perspectives in Bioanalysis*, 2:3–44, 2007.

[2] R. A. Soo, L. Z. Wang, S. S. Ng, P. Y. Chong, W. P. Yong, S. C. Lee, J. J. Liu, T. B. Choo, L. S. Tham, H. S. Lee, B. C. Goh and R. Soong. Distribution of gemcitabine pathway genotypes in ethnic Asians and their association with outcome in non-small cell lung cancer patients. *Lung Cancer*, 63(1):121–127, 2009.

[3] 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)*, 34(1):94–99, 2008.

[4] J. Zhang, R. Chiodini, A. Badr and G. Zhang. The impact of next-generation sequencing on genomics. *Journal of Genetics and Genomics*, 38(3):95–109, 2011.

[5] J. M. Corchado, J. F. De Paz, S. Rodríguez and J. Bajo. Model of experts for decision support in the diagnosis of leukemia patients. *Artificial Intelligence in Medicine*, 46(3):179–200, 2009.

[6] F. Gregoretti, G. Laccetti, A. Murli, G. Oliva and U. Scafuri. MGF: A grid-enabled MPI library. *Future Generation Computer Systems*, 24(2):158–165, 2008.

[7] H. Jin, D. Jespersen, P. Mehrotra, R. Biswas, L. Huang and B. Chapman. High performance computing using MPI and OpenMP on multi-core parallel systems. *Parallel Computing*, 37(9):562–575, 2011.

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

[9] M. Wooldridge. Introduction to MultiAgent Systems. John Wiley and Sons, 2002.

[10] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004.

[11] S. S. Sahua and G. Pand. Identification of Protein-Coding Regions DNA Sequences Using A Time-Frequency Filtering Approach. *Genomics, Proteomics and Bioinformatics*, 9(1-2):45–55, 2011.

[12] A. Khajeh-Saeed, S. Poole and J. Blair. Acceleration of the Smith-Waterman algorithm using single and multiple graphics processors. *Journal of Computational Physics*, 229(11):4247–4258, 2010.

[13] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[14] W. J. Kent. BLAT-the BLAST-like alignment tool. *Genome Research*, 12:656–664 2002.

[15] 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 et al. Genome se-quencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376-80, 2005.

[16] J. R. Miller, S. Koren and G. Sutton. Assembly algorithms for next-generation sequencing data. *Genomics*, 95(6):315–327 2010.

[17] B. Delaney, J. D. Astwood, H. Cunny, R. E. Conn, C. Herouet-Guicheney, S. MacIntosh, L. S. Meyer, L. Privalle, Y. Gao, J. Mattsson and M. Levine. ILSI International Food Biotechnology Committee Task Force on Protein Safety. Evaluation of protein safety in the context of agricultural biotechnology. *Food and Chemical Toxicology*, 46:S71–S97, 2008.

[18] S. W. Blume, N. L. Jackson, A. R. Frost, W. E. Grizzle, O. D. Shcherbakov, H. Choi and Z. 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*, 88(3):341–352, 2010.

[19] M. Wooldridge and N. R. Jennings. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[20] N. David, M. B. Marietto, J. S. Sichman and H. Coelho. The Structure and Logic of Interdisciplinary. Research in Agent-Based Social Simulation. *Journal of Artificial Societies and Social Simulation*, 7(3), 2004.

[21] T. Schelling. Micromotives and macrobehavior. New York: W. W. Norton, 1978.

[22] J. M. Epstein and R. L. Axtell. Growing Artificial Societies: Social science from the bottom up. MIT Press, 224pp, 1996.

[23] E. Oliva, A. Natali, A. Ricci and M. Viroli. An Adaptation Logic Framework for Java-based Component Systems. *Journal of Universal Computer Science*, 14(13):2158–2181, 2008.

[24] M. O. Shafiq, Y. Ding and D. Fensel. Bridging Multi Agent Systems and Web Services: towards interoperability between Software Agents and Semantic Web Services, *In* Proceedings of the 10th IEEE International Enterprise Distributed Object Computing Conference (EDOC'06). IEEE Computer Society, pages 85–96, 2006.

[25] X. Liu. A Multi-Agent-Based Service-Oriented Architecture for Inter-Enterprise Cooperation System. *In* Proceedings of the Second international Conference on Digital Telecommunications (ICDT'07). IEEE Computer Society, 2007.

[26] P. Rigole, T. Holvoet and Y. Berbers. Using Jini to Integrate Home Automation in a Distributed Software-System. *In* 4th International Workshop on Distributed Communities on the Web, LNCS 2468, pages 291–303, 2002.

[27] C. Walton. Agency and the Semantic Web. Oxford University Press, 272pp, 2006.

[28] G. Billiau, C. F. Chang, A. Miller and A. Ghose. Support-based distributed optimisation: an approach to radiotherapy patient scheduling. Studies in Health Technology and Informatics, 159:229–233, 2010.

[29] L. Vignal and F. Lisacek Ghose. A Multi-Agent System for Exon Prediction in Human Sequences. *In* Eighth Workshop on Genome Informatics, pages 156–165, 1997.

[30] C. G. Ralha, H. W. Schneider, L. O. Fonseca, M. E. Walter and M. M. Brigido. Using BioAgents for Supporting Manual Annotation on Genome Sequencing Projects. *In* Advances in Bioinformatics and Computational Biology, LNCS 5167, pages 127–139, 2008.

[31] K. Decker, X. Zheng and C. Schmidt. A multi-agent system for automated genomic annotation. *In* Proceedings of the fifth international conference on Autonomous agents, pages 433–440, 2001.

[32] F. Tataria, M. R. Akbarzadeh-T and A. Sabahi. Fuzzy-probabilistic multiagent system for breast cancer risk assessment and insurance premium assignment. *Journal of Biomedical Informatics*, DOI:10.1016/j.jbi.2012.05.004.

[33] W. Jiang, M. Baumgarten, Q. Dai and Y. Zhou. The deployment and evaluation of a bioinformatics grid platform - The HUST_Bio_Grid. *Computers and Electrical Engineering*, 38(1):19–34, 2012.

[34] J. Wu, X. Xu, P. Zhang and C. Liu. A novel multi-agent reinforcement learning approach for job scheduling in Gridcomputing. *Future Generation Computer Systems*, 27(5):430–439, 2011.