A Simple Algorithm Implementation for Pattern-Matching with Bounded Gaps in Genomic and Proteomic Sequences, on the Grid EGEE Platform, using an intuitive User Interface

Vegoudakis K. I., Margaritis K. G., Maglaveras N.

Abstract In the last decade an unprecedented development in bioinformatics has been observed. An extremely high number of organisms have been sequenced and included in genomic databases. The huge amount of data produced needs to be stored and processed for further analysis. Scientists, have researched algorithms for finding complicated patterns in DNA sequences, but there is a need for computational power and large storage systems, in order to implement specific algorithms in as many as possible DNA sequences stored in large Databases and save the results produced for future use.

Recently, a number of pattern matching algorithms that allowing gaps have been introduced and Grid is an emerging techology that seems to be helpful in this kind of biomedical research. In this paper, we present our effort towards the construction of a user friendly Interface for accessing the Grid EGEE platform. The Interface is a specific one and was built to perform Pattern-Matching with Bounded Gaps in Genomic and Proteomic Sequences. The algorithm and the Interface is tested with small and large DNA and protein sequences as an input, downloaded from a gene and protein repository, NCBI (National Center for Biotechnology Information) and the gathered results produced are presented.

Vegoudakis Konstantinos
Parallel and Distributed Processing Laboratory,
Department of Applied Informatics, University of Macedonia, 156 Egnatia str., P.O. Box 1591, 54006 Thessaloniki, Greece, e-mail: kostasve@uom.gr

Margaritis Konstantinos
Parallel and Distributed Processing Laboratory,
Department of Applied Informatics, University of Macedonia, 156 Egnatia str., P.O. Box 1591, 54006 Thessaloniki, Greece, e-mail: kmarg@uom.gr

Maglaveras Nikos
Lab of Medical Informatics, Health Sciences Faculty, Aristotle University, AUTH PB 323, 54124 Thessaloniki, Greece, e-mail: nicmag@med.auth.gr

1 Introduction

Bioinformatics is playing an increasingly large role in the study of fundamental biomedical problems and represents a new, growing area of science that uses computational approaches to answer biological questions [1]. The explosion of sequence and structural information available to researchers need to be processed and stored in special supercomputing infrastructures and storage systems respectively. Grid is an emerging technology for distributed computing in advanced science and engineering and was firstly developed as a concept to enable resource sharing within scientific collaborations [2]. Although, great investments have been made the previous years worlwide towards the constuction of Grid Infrastructures, the Grid community expects to evaluate the necessity and the current potential of using the Grid [3].

Grid technologies have enormous potential for heavy computational and storage demanding applications but a few of them have been implemented and executed in the context of Grid computing. There are a lot of reasons of the slow take up of Grid computing but the most substantial one is the lack of an existing simple unique framework and easy to use User Interface for executing applications either biomedical or not. For example the glite WMS (Workload Management System) demands the experience of using Unix-like scripts in order to interact with the Grid infrastructure.

The potential of Grid computing in healthcare has been examined and worked out by the HealthGrid initiative [4] according to which the prospects of Grid Computing are deployed e.g. computational models of systems/organs, pharmaceutical research, large-scale epidemiological studies and so on. The scale and the complexity of the Euroeopan EGEE (Enabling Grid for E-sciencE) project and the launch and the expanding of biomedical applications in it, is described in [5]. It is worth mentioning the BIOMED Virtual Organization (VO) which is hosted on the EGEE project and created in the context of the BioinfoGRID project, whose purpose is to promote the Bioinformatics applications for life science, in order to carry out research based on the Grid networking technology [6].

User-friendly access to the grid environment is of great importance for the scientific community. In particular a lot of effort has been made to construct an interface which is easy to use. The g-Eclipse project aims to build an integrated workbench framework to access the power of existing Grid infrastructures. g-Eclipse tries to ease the execution of the existing applications that are needed to be executed in the Grid environment and it provides tools for the customization of the Grid users' applications and management of the Grid resources [7]. GuiGen is a comprehensive set of tools for creating customized graphical user interfaces and was originally designed for the use in computational grids [8]. In addition, P-GRADE provides a high-level graphical environment to develop parallel applications transparently both for parallel systems and the Grid and it also supports workflow definition and coordinated multi-job execution for the Grid. One of the main advantages of P-GRADE is that the user does not have to learn the different APIs for parallel systems and the Grid. The current version of P-GRADE supports the interactive execution of parallel programs both on Globus-2 and Condor Grids [9]. However, the above mentioned approaches still require technical knowledge from the end users.

In this work, which is a sequence of previous work on the research of an intuitive user-friendly and generic User Interface [10], [11], either with the help of XML or not, another user-friendly interface for Pattern-Matching with Bounded Gaps in genomic sequences is presented. This inteface is a specific one and it is used only for submission of certain type of jobs, those for string pattern matching with bounded gaps. The user has the ability to interact with the UI for the job submission, job status retrieval, upload/download of DNA and protein sequence files that are needed, etc.

The problem of fast searching of patterns that contain Classes of characters and Bounded size Gaps (CBG) in text occurs in various fields and the most important one, is protein matching. The design of two new practical CBG practical algorithms that are faster and simpler than all regular expression search techniques are described in [12]. In Crochemore et. al. [13], algorithms for several versions of approximate string pattern mathching with gaps are presented. Further restrictions to the gaps are introduced in [14] with lower and upper bounds restrictions on the gaps. The user interface created, implements these algorithms on the Grid platform.

The applicability of the user interface is examined via a set of jobs. Different DNA and Protein sequences were used to search in them for different patterns with bounded gaps. It has to be noted that especially large DNA sequences need a lot of computing time and storage in order to execute. For this reason the interface built, enables the user to submit multiple executions of jobs with string pattern matching with bounded gaps. This method is known as parameter study and it is often met in biomedical applications.

More precisely, the user interface is expected to utilize existing bioinformatics applications on available grid testbed (such as EGEE, NGS, etc), [15]. In its final form it will require only Java. As a result installation and configuration of specific operating system and grid middleware toolkit will not be necessary. The user interface current production status, simplifies the job submission process on EGEE grid infrastructure, which is not a trivial task from a biologist's end-user point of view. The parameter study as mentioned above, gives the biologist an assistance in his work, as he gains time, retrieving all the results from the experiments in a reasonable amount of time.

This paper is structured as follows. First, the methodology adopted is presented, providing the necessary implementation and introductory details about the string pattern matching problem with bounded gaps and the creation of the UI for the Grid. Then, the results obtained from the execution of multiple jobs on the Grid environment, are presented. Finally, our future research perspectives and the conclusions drawn are discussed.

2 Methodology

2.1 Basic Definition of the String Matching Problem with Bounded Gaps

The adoption of two uniformly Alphabets [14] $\Sigma_{DNA} = \{A, C, G, T\}$ and $\Sigma_{Protein} = \{A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$ is needed in order to define the string pattern matching problem with bounded gaps. Each letter is standing for the first letter of the chemical name of the nucleotide in a DNA sequence and each letter in the protein alphabet represents the amino acid abbreviation for each protein in a protein sequence.

Let X be a string drawn from Σ_{DNA}. X represents an array X[1..n] of n$\geq$ 0 symbols, where n=length(X) denotes the lentgh of the string X. X[i] denotes the ith symbol of X. In addition, X[i..j] denotes the substring of X contained between ith and the jth symbol. Given a text T of length n and a pattern P of length m, an occurence with b-bounded gaps of P in T is an increasing sequence of indices $i_1, i_2, ..., i_m$ such that (i) $1 \leq i_1$ and $i_m = i \leq n$ and (ii) $i_{h+1} - i_h \leq \mathtt{b} + 1$, for h=1,2,...,m-1. $P \trianglelefteq^i_b T$ means that P has an occurence with b-bounded gaps that terminates at position i in text T. In the same way, an occurence with a-bounded gaps of P in T is an increasing sequence of indices $i_1, i_2, ..., i_m$ such that (i) $1 \leq i_1$ and $i_m = i \leq n$ and (ii) $i_{h+1} - i_h \geq \mathtt{a} + 1$, for h=1,2,...,m-1. $P \trianglelefteq^i_a T$ means that P has an occurence with a-bounded gaps that terminates at position i in text T. Finally, an occurence with (a,b)-bounded gaps of P in T is an increasing sequence of indices $i_1, i_2, ..., i_m$ such that (i) $1 \leq i_1$ and $i_m = i \leq n$ and (ii) $\mathtt{a} + 1 \leq i_{h+1} - i_h \leq \mathtt{b} + 1$, for h=1,2,...,m-1. $P \trianglelefteq^i_{(a,b)} T$ means that P has an occurence with (a,b)-bounded gaps that terminates at position i in text T.

In [14], four algorithms are described. The interface for the Grid implements specifically the string pattern matching problem with b - bounded gaps, a-bounded gaps and (a,b)-bounded gaps. The three implemented algorithms for the solution of the string pattern matching problems with bounded gaps are described as follows:

Problem 1 (upper bounded gaps) Given a text T of length n, a pattern P of length m and a positive integer b, the STRING PATTERN MATCHING PROBLEM WITH b-BOUNDED GAPS is to find all positions j in T such that $P \trianglelefteq^i_b T$, for $1 \leq j \leq n$.

Problem 2 (lower bounded gaps) Given a text T of length n, a pattern P of length m and a positive integer a, the STRING PATTERN MATCHING PROBLEM WITH a-BOUNDED GAPS is to find all positions j in T such that $P \trianglelefteq^i_a T$, for $1 \leq j \leq n$.

Problem 3 (lower & upper bounded gaps) Given a text T of length n, a pattern P of length m and positive integers (a,b), the STRING PATTERN MATCHING PROBLEM WITH (a,b)-BOUNDED GAPS is to find all positions j in T such that $P \trianglelefteq^i_{(a,b)} T$, for $1 \leq j \leq n$.

In addition, the fourth algorithm mentioned in [14] is not implemented due to its nature of expecting many parameter, that is, `a` and/or `b` restrictions on every possible gap between every nucleotide. The specific interface could be expanded with Regular Expressions or with the help of a special parser for the input of the fourth algorithm. An other feature Regular Expressions can provide, is finding occurences of patterns with bounded gaps using the IUPAC nucleotide code.

In other words, from a biologist's point of view, Problems 1, 2 and 3 find all positions in a DNA or protein sequence T with at most `b` gaps, at least `a` gaps and between `a and b` gaps respectively, between every two nucleptides in the pattern specified. Setting `a`=0 and `b`=0 turns the problem in finding a specific pattern in a DNA sequence. The problems described above need plenty of time to run with large DNA sequences in length. For this reason, two complementary options are examined: Grid computing which offers the ability to run computationally and storage intensive applications [15] and a GUI friendly enough, in order to submit multiple instances of the algorithm with different `a` and/or `b`. As a result, a biologist can benefit from the fact of retrieving his/her result in a reasonable amount of time.

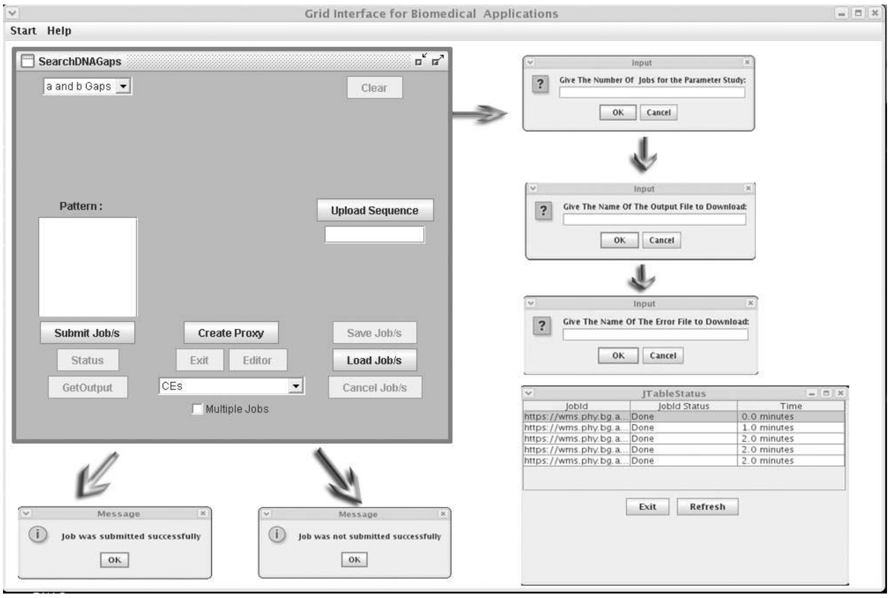
2.2 The Implementation of the User Interface

The GUI in Fig. 1 incorporates all the necessary steps for submitting and managing a job in a Grid environment. The control panel (Submit job/s, Status, Create Proxy, Save Job/s, Load Job/s, Cancel Job/s) which offers a user-friendly job management, also exists in the specific User Interface designed for string pattern matching with Bounded gaps.

EGEE was the Grid infrastructure that our interface utilized and gLite 3.1 [16] was the necessary middleware for accessing the Grid platform. The GUI was developed in Java and WMProxy API (Application Programming Interface) was used for submitting, cancelling and retrieving the output. WMProxy is implemented as a web service. A web service allows us to take advantage of the benefits of the web, not only to provide information, but also to offer services to a greater community of possible users [17]. Within the bioinformatics community, an average end-user might need to access and use hundreds of databases and tools on a given day [18].

The creation of a VOMS (Virtual Organization Membership Service) proxy certificate for accessing the Grid and the status of the submitted jobs are handled by the gLite user interface via the use of java. For data uploading the gsiFTP client [11] was used for uploading the C++ implementation of the string pattern matching with bounded gaps algorithm.

All the scripts and the JDL (Job Description Language) files that are needed for the submission of the jobs are generated automatically via the GUI. This automization saves time and makes the submission of jobs for the naive user simpler enough. The only parameters the user has to fill in, are the number of the jobs, if he wants to perform a parameter study and then the names of the error and output files. Through



Give The Number Of Jobs for the Parameter Study:

Give The Name Of The Output File to Download:

Give The Name Of The Error File to Download:

JobId	JobId Status	Time
https://wms.phy.bg.a...	Done	0.0 minutes
https://wms.phy.bg.a...	Done	1.0 minutes
https://wms.phy.bg.a...	Done	2.0 minutes
https://wms.phy.bg.a...	Done	2.0 minutes
https://wms.phy.bg.a...	Done	2.0 minutes

Job was submitted successfully

Job was not submitted successfully

Fig. 1 The specific User Interface, SearchDNAGaps that implements and eases the string pattern matching with bounded Gaps

a list box, the user has also the abilty to choose from a list of predifined CEs (Computing Element), or a random one.

The uploading of the DNA or protein sequences in fasta format is performed, via the use of the GUI. It has to be mentioned that each fasta file should contain only one DNA sequence as an input, because in this implementation different patterns are utilized and searched for occurrences in the DNA sequence. The user is enabled to choose one of the three problems for string pattern matching with bounded gaps mentioned in Sect. 2.1.

An another important element of the specific user interface, is the submission and the management of multiple jobs (Status, Save, Load, Cancel, Get ouput).As a result, the parameter study for different values of the string pattern matching algorithm with bounded gaps (`a,b, a and b`) can be performed and retrieval of output is faster and efficient.

3 Test Case and Results

Our algorithm implementation was tested on the Grid EGEE platform with different DNA sequences that differ in size, downloaded randomly from NCBI. Figure 2[1], illustrates these results. Twelve different DNA sequences ranged from 3626 to

[1] Arabidopsis2 and Arabidopsis3 are segments of the Arabidopsis DNA sequence

30.423.563 base pairs and one small Pattern, ATGCGCG, were used as an input. For each DNA sequence, a, b, and (a,b) parameters were initilized with the same values five times. The values of a,b were chosen randomly, according to the three problems described in Sect. 2.1. The results are illustrated in Fig. 2 and one can easily draw the conclusion that the execution time of a job for a string pattern matching with bounded gaps depends on the length of the DNA sequences. More precisely, by keepping the pattern unchanged, if the length of a DNA sequence gets over the eigth million base pairs, then the execution time grows rapidly. For this reason, Grid can be used with multiple job submission and different parameter values, with large DNA sequences in length. Also, the small deviation in execution times for different a and/or b is due to the fact that CEs were chosen randomly.

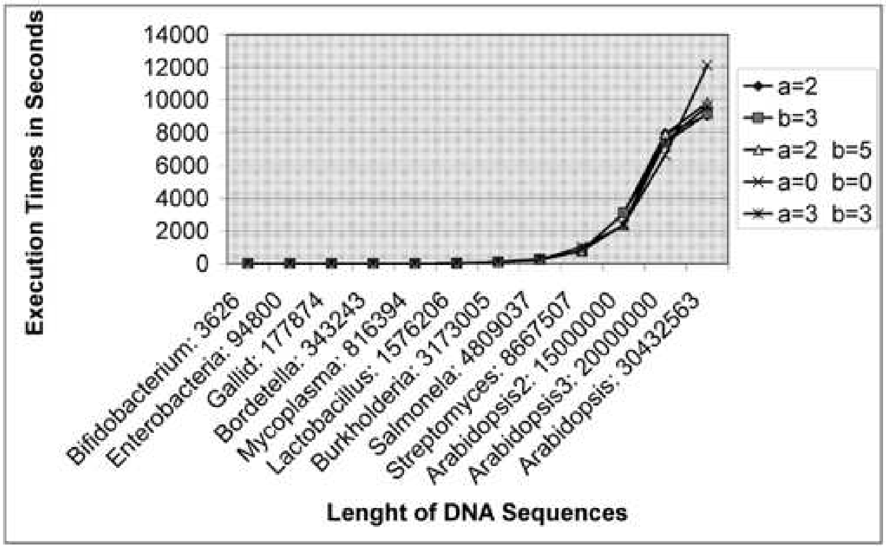


Fig. 2 Execution Times in seconds of the string pattern matching algorithm with bounded gaps for different DNA sequences and (a, b, a and b) parameters

As far as Protein sequences concerned, it has to be noted that they are rather small in length. Titin protein with gi number (genInfo identifier), gi|108861911 was used for testing. The results produced were as they were expected to be. Due to the small sizes of protein sequences, the execution times are rather small, only five seconds. Today, personal computers are very fast and they can handle small DNA or protein sequences efficiently as Fig. 2 shows. The use of Grid is strongly recomended for large sequences, that require hours for the execution of a simple job of string pattern matching.

4 Discussion - Conclusions

In this paper, a specific user-friendly interface was developped for the execution of specific jobs, those for string pattern matching with bounded gaps in DNA or Protein sequences. The rationale of this work is to present the Grid infrastrucure as a mean, that can execute computationally and storage exhausive applications, in the concept of a parameter study approach, by using the multible job submission feature of the proposed specific GUI. Table 1 summarizes and gives a brief explanation of each of the technical terms, projects, and resources appeared in the paper.

The applicability and the potentiality of the proposed specific GUI was illustrated by testing it with different DNA sequences that vary in the size of length. Protein sequences are rather small in size and the algorithm needs only seconds to be executed, so personal computers can be used for this kind of string pattern matching.

The implementation of the algorithm takes as input only one sequence at each job and examines for bounded gaps. In the future, the algorithm and the Grid interface will be changed to query a large number of different sequences saved in fasta format, at each job. Additionally, the combination of the string pattern matching with bounded gaps algorithm, with approximate string matching algorithms [19], will allow to find the positions of a text where a given patterns occurs, allowing a number of ”errors” in the matches with bounded gaps. The parallelization of the algorithm with MPI (Message Passing Interface) and the use of multiple submission or job workflows could expand the features of the GUI, facilitating the user to submit fewer jobs.

The current implementation requires the installation of the gLite middleware in the computer hosting the GUI. This limitation is expected to be solved in the near future and we working on that. Finally, the need of a mobile Graphical User Interface for the Grid using the Java techologies, is in our nearest expectations and we are looking to present one in the nearest future.

Table 1 A brief explanation of each of the technical terms, projects, and resources appeared in the paper

Term	Brief Explanation	URL resource
gLite WMS	gLite Workload Management System	http://glite.web.cern.ch/glite/packages/R3.1/deployment/glite-WMS/glite-WMS.asp
gLite UI	gLite User Interface	http://glite.web.cern.ch/glite/packages/R3.1/deployment/glite-UI/glite-UI.asp
EGEE	Enabling Grids for E-sciencE	http://www.eu-egee.org/
NGS	The National Grid Service	http://www.grid-support.ac.uk/
NCBI	The National Center for Biotechnology Information	http://www.ncbi.nlm.nih.gov/
HealthGrid	HealthGrid Community	http://community.healthgrid.org/
BioinfoGRID	The BioinfoGRID project	http://www.bioinfogrid.eu/
g-Eclipse	g-Eclipse project	http://www.geclipse.org/
GuiGen	GuiGen is a comprehensive set of tools for creating customized graphical user interfaces (GUIs)	http://www.zib.de/schintke/guigen/ index.en.html
P-GRADE	Parallel Grid Run-time and Application Development Environment	http://www.p-grade.hu/
API	Application programming interface	http://en.wikipedia.org/wiki/API
Globus Toolkit	The Globus Toolkit is an open source software toolkit used for building Grid systems and applications	http://www.globus.org/
Condor Project	The Condor Project	http://www.cs.wisc.edu/condor/
CBG algorithms	Classes of characters and Bounded size Gaps algorithms	Navarro G, Raffinot M (2001)
Regular Expressions	Regexp (for short) is a special text string for describing a search pattern	http://java.sun.com/docs/books/tutorial/essential/regex/
IUPAC	International Union of Pure and Applied Chemistry	http://www.bioinformatics.org/sms/ iupac.html
GUI	Graphical User Interface	http://infovis.cs.vt.edu/GUI/java/
WMProxy	WMProxy is a new component to access the gLite Workload Management System (WMS)	http://trinity.datamat.it/projects/EGEE/wiki/wiki.php?n=WMProxyService.AboutWMProxyService
VOMS	Virtual Organization Membership Service	http://www.globus.org/grid_software/security/voms.php
gsiFTP client	A client developed by using GridFTP (GSI enabled FTP) protocol for the file transfers	http://www-unix.globus.org/cog/distribution/1.1/API.html
JDL	Job Description Language	https://edms.cern.ch/file/590869/1/EGEE-JRA1-TEC-590869-JDL-Attributes-v0-8.pdf
MPI	The Message Passing Interface standard	http://www.mcs.anl.gov/research/projects/mpi/

References

1. Baxevanis A. D. , Francis Ouellette B. F.: Bioinformatics and the Internet. In: BIOINFORMATICS: A Practical Guide to the Analysis of Genes and proteins. SECOND EDITION. pp 1-17 John Wiley & Sons, New York, (2001)
2. Foster I., Kesselman C.: Concepts and Architecture. In: The Grid: Blueprint for a New Computing Infrastructure. Elsevier, San Francisco, (2004)
3. Goble C.: The Grid needs you! Enlist now. In: On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE, ser. LNCS, R. Meersman et al. Eds. Berlin Heidelberg: *Springer-Verlag* , vol. 2888, pp. 589 - 600, (2003)
4. Breton V et al (2005) The Healthgrid white paper, HealthGrid Proc., Oxford, UK, pp 249 - 321
5. Gagliardi F. et al.: Building an infrastructure for scientific Grid computing: status and goals of the EGEE project, Phil. Trans. R. Soc. A (2005) doi:10.1098/rsta.2005.1603
6. Milanesi L et al (2007) White paper: guidelines and recommendations for the scientific community based on the experience and the results gained from the BioinfoGRID project, http://www.bioinfogird.eu
7. g-Eclipse at http://www.geclipse.eu/
8. Reinefeld A, Stuben H, Schintke F, Din G (2002) GuiGen: A toolset for creating customized interfaces for grid user communities. Future Generation Computer Systems 18(8): 1075 - 1084
9. Kacsuk P., Dozsa G., Kovacs J., Lovas R., Podhorszki N., Balaton Z. and Gombas G. (2003) P-GRADE: A Grid Programming Environment. Journal of Grid Computing 1: 171197
10. Vegoudakis K., Koutkias V., Malousi A., Chouvarda I. and Maglaveras N. A generic Grid interface and execution framework for biomedical applications. BioInformatics and BioEngineering, BIBE 2008. 8th IEEE International Conference.
11. Vegoudakis K., Koutkias V., Malousi A., Chouvarda I. and Maglaveras N. Towards User-friendly Interfacing of Biomedical Applications with the Grid: A Paradigm with SVM Optimization for Gene Prediction, in Proc. of the 4th European Congress for Medical and Biomedical Engineering 2008 (eMBEC2008), 23-27 November, Antwerp, 2008.
12. Navarro G, Raffinot M (2001) Fast and simple character classes and bounded gaps pattern matching, with application to protein searching. Proceedings of the fifth annual international conference on Computational biology, Montreal, Quebec, Canada Pages: 231 - 240
13. Crochemore M, Makris C, Rytter W, Tsakalidis A, Tsichlas K (2002) Approximate String Matching with Gaps. Nordic Journal of computing, 9(2002): 54 - 65
14. Pinzon YJ, Wang S (2005) Simple algorithm for pattern-matching with bounded gaps in genomic sequences. In Proceedings of ICNAAM05, pages 827 - 831
15. Kransogor N., Shah A.A., Barthel D., Lukasiak P. and Blazewicz J., January 2008. Web and Grid Technologies in Bioinformatics, Computational and Systems Biology: A Review. Current Bioinformatics, 3, pp. 10 - 31(22)
16. gLite documentation at http://glite.web.cern.ch/glite/documentation/
17. Avellino G. et al, 2006. Flexible job submission using web services: The gLite WMProxy experince. 15th International Conference on Computing In High Energy and Nuclear Physics, Mumbai, India, pp.831 - 835
18. Curcin V., Ghanem M. and Guo Y., 2005. Web services in the life sciences. Drug discovery today, 10(12), pp. 865-871.
19. Navarro G., 2001. A Guided Tour to Approximate String Matching. ACM Computing Surveys, 33(1), pp. 31 - 88