World Applied Sciences Journal 2 (6): 582-586, 2007
ISSN 1818-4952
© IDOSI Publications, 2007

Microorganism DNA Pattern Search in a Multi-agent Genomic Engine Framework

Mohebbat Mohebbi, R. Mohammad, T. Akbarzadeh and Amin Milani Fard

[1]Department of Food Science and Technology, Ferdowsi University of Mashhad, Iran
[2]Department of Electrical Engineering, Ferdowsi University of Mashhad, Iran
[3]Department of Computer Engineering, Ferdowsi University of Mashhad, Iran

Abstract: Considering growth of genomic and proteomic data, application of multi-agent information gathering technologies has become a fertile domain. In this research a multi-agent system is designed for distributed DNA pattern search which can be regarded as a solution to the challenges of post genomic era. This model can contribute to faster prototyping of information gathering systems in the heterogeneous web environment with dynamic database.

Key words: Multi-agent system · genomic · pattern search · DNA · Microorganism

INTRODUCTION

Bioinformatics is commonly defined as the application of information science and technology to the management of biomedical data and information and it has concerned itself with storage, management and analysis of biologically relevant data derived from experimental biological analysis. While initially bioinformatics involved predominantly sequence information along with hand curated annotion, now it covers a huge amount of data that are stored, used and manipulated over the Internet.

Bio-search engines and portals aim to retrieve more accuracte and relevant information using advanced methods. Sharing a high amount of data which is often raw, heterogeneous and distributed over the web and changes continuously, is one of the main problems which necessitates application of novel information retrieval science and distributed systems in bioinformatics. Considering large amount of dynamically changing distributed data, multi-agent systems are attractive appraches for solving such problems [1].

In this paper, we have developed a multi-agent system for DNA pattern search and applied it to the real and demanding problems of genomic analysis. This solution gives us mechanisms for dealing with changing data and the dynamic appearance of new sources.

RELATED WORKS

Many molecular biology problems on sequences can be formulated as string matching problems including:

- Storing, retrieving and comparing DNA strings
- Comparing two or more strings for similarities and
- Searching databases for related strings and substrings
- Determining the physical and genetic maps
- Looking for structural patterns in DNA and RNA
- And much more...

DNA is a sequence of base-pairs of the four nucleotide bases Adenosine, Cytosine, Thymine and Guanine (A, C, T and G). In fact it must be notified that one of the most important issues in molecular biology and genetics is reading DNA sequences. Despite the impressive achievements in sequencing whole genomes, there is still a need for an improvement of the existing sequencing methods or for a development of the new ones regarding computation point of view [2].

Finding similar regions common to each sequence in a given set of DNA, RNA, or protein sequences is a necessity in many problems in molecular biology. These problems find applications in locating binding sites and finding conserved regions in unaligned sequences, in genetic drug target identification, in designing genetic probes, in universal PCR primer design and, outside computational biology, in coding theory [2-4].

The Semantic Web community aims at technological steps of performing semantic access and there is a clear justification of developing a semantic web for genomic data. The Gene Ontology Consortium [5] and TAMBIS [6], amongst others have sought to formally describe biological knowledge in the new OWL (Ontology Web

Corresponding Author: Dr. Mohebbat Mohebbi, Department of Food Science and Technology, Ferdowsi University of Mashhad, Iran

Language) the w3c [7] adopted standard for formal ontology markup on the web. Semantic access is achieved through ontologies associated with web sites.

Ontologies describe the information in a way that is precise and formal enough to be manipulated by reasoning software and query tools. Agents are integral to the semantic web: An agent is software that knows how to navigate the web and to use the available knowledge to achieve a given task. In order to achieve its task an agent may need to cooperate with other agents, it may need to learn more and it may need to be proactive in response to changing data as the data becomes available on the web. The Fungal Web Ontology, written in OWL, covers enzyme classification, enzyme functional parameters, enzyme applications and fungal taxonomy gathered from distributed resources [8, 9].

PROPOSED SYSTEM ARCHITECTURE

An important class of problems in molecular biology includes DNA pattern search. Genomic and proteomic sequence databases provide a fundamental reference tool for molecular biologists in which searching databases for DNA sequences similar to a query sequence, even to compare whole genomes and to identify sequences of unknown origin or functions is a key issue. In this research a multi-agent framework is designed to support the definition of distributed application in a biological domain. The project is implemented using *Java Agent DEelopment* (JADE) framework [10] and *MySql* database management system.

JADE is a software development framework aimed at developing multi-agent systems and applications in which agents communicate using FIPA Agent Communication Language (ACL) messages and live in containers which may be distributed to several different machines. The Agent Management System (AMS) is the agent who exerts supervisory control over access to and use of the Agent Platform. Only one AMS will exist in a single platform. Each agent must register with an AMS in order to get a valid AID. The Directory Facilitator (DF) is the agent who provides the default yellow page service in the platform. The Message Transport System, also called Agent Communication Channel (ACC), is the software component controlling all the exchange of messages within the platform, including messages to/from remote platforms. The standard model of an agent platform, as defined by FIPA, is represented in the Fig. 1.

JADE is capable of linking Web services and agents together to enable semantic web applications. A Web service can be published as a JADE agent service and an agent service can be symmetrically published as a Web service endpoint. Invoking a Web service is just like invoking a normal agent service. Web services' clients can also search for and invoke agent services hosted within JADE containers. The Web Services Integration Gateway (WSIG) [11] uses a Gateway agent to control the gateway from within a JADE container. Interaction among agents on different platforms is achieved through the ACC.

Whenever a JADE agent sends a message and the receiver lives on a different agent platform, a Message Transport Protocol (MTP) is used to implement lower level message delivery procedures [12]. There are two main MTPs available today to support this inter-platform

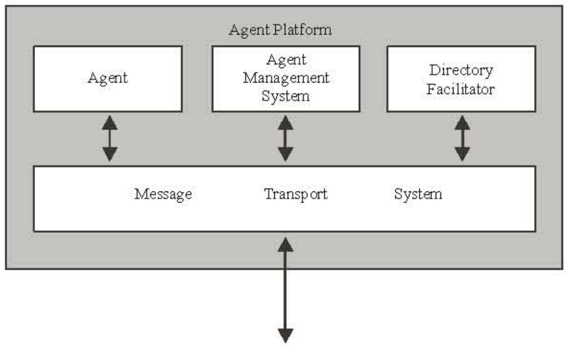

Fig. 1: Reference architecture of a FIPA Agent Platform

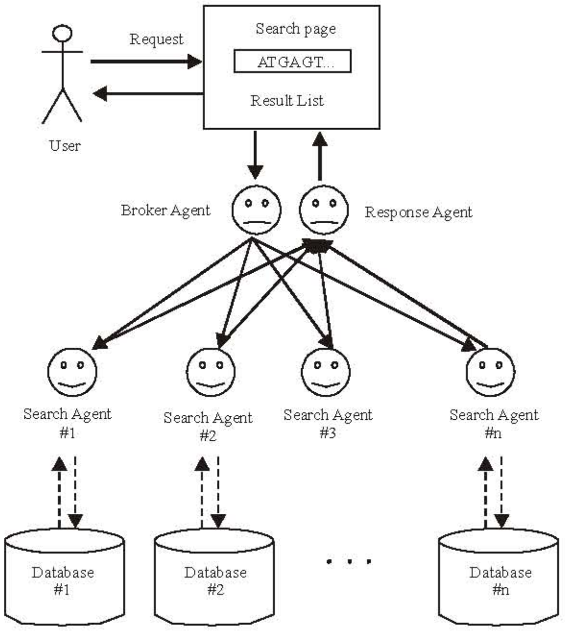

Fig. 2: The proposed system architecture

```
KMP FAILURE (P)
Input: Pattern with m characters
Output: Failure function f for P[i..j]
i ← 1
j ← 0
f(0) ← 0
while i < m do
  if P[j] = P[i]
    f(i) ← j +1
    i ← i +1
    j← j + 1
else if
    j ← f(j - 1)
else
    f(i) ← 0
    i ← i +1
```

```
KMP (T, P)
Input: Strings T[0..n]& P[0.. m]
Output: Starting index of substring of T
matching P
f ← compute failure function of Pattern P
i ← 0
j ← 0
while i < length[T] do
  if j ← m-1 then
    return i- m+1 // a match
i ← i +1
j ← j +1
else if j > 0
    j ← f(j -1)
  else
    i ← i +1
```

Fig. 3: The KMP algorithm

agent communication-CORBA IIOP-based and HTTP-based MTP. Considering large-scale applications over separated networks, agent communications has to be handled behind firewalls and Network Address Translators (NATs), however, the current JADE MTP do not allow agent communication through firewalls and NATs. Fortunately, the firewall/NAT issue can be solved by using the current JXTA implementation for agent communication [13].

JXTA is a set of open protocols for P2P networking. The JXTA protocols enable developers to build and deploy P2P applications through a unified medium [14]. A JXTA peer is any networked device that implements one or more of the JXTA protocols. Several peers can self-organize into peer groups, which provide a common set of services. Pipes are bound to specific endpoints at runtime, such as a TCP port and an associated IP address. The supported objects for transmission are binary code, data strings and Java-based objects. JXTA peers advertise their services via advertisements, which enable other peers on the network to learn how to connect to and interact with peer's services. The advertisements are XML-based documents composed of a series of hierarchically arranged elements.

The JXTA's Peer Discovery Protocol (PDP) is used to discover any published resources, both on client and server sides. Obviously, JXTA is a suitable architecture for implementing MTP-s for JADE and consequently JADE agent communication within different networks can be facilitated by incorporating JXTA technology into JADE. The performance comparison of the JXTA MTP with IIOP and HTTP MTPs showed that performance of the JXTA implementation is similar to the IIOP and HTTP implementation and is even better, if the number of participating agents exceeds 100. This means that the scalability of JXTA implementation is good enough to be applied in large-scale multi-agent systems [13].

Regarding agent communication method via JXTA in different networks [13], we proposed our MAS architecture in a local network using JADE which is to some extend a simulation for real demand web interaction process. In this work the user gives a query to the search agent and search agents looks for the answer using the terms in the ontologies available. Agents may also communicate with other agents and exchange information to broaden the search results. Our proposed architecture uses different types of agent and each has its own characteristics.

Broker agent: The Broker Agent (BA) after getting the query request does a simple task sharing process. BA can also create search agents if needed.

Search agents: These agents will return the search result to the response agent according to their relative field and database.

Response agent: This agent has the responsibility to show the result of retrieved information. To do so Response Agent (RA) collects SAs results, make a ranking out of them and then write them on screen ordered by relation percentage. RAs use fuzzy approach to select better result upon the queries.

The basic text processing part of our work is pattern matching in which we are given a substribng DNA pattern of length *m* and a database with *r* records each of which containing text of length at most *n* and we wish to know number of pattern *p* occurance in the record *r*. Several algorithms have been developed for pattern matching, however, we take advantage of a well-known linear time

Table 1: Results of sample user query

Rank	Microorganism class	#	Rank	Microorganism class	#
1	Coccidioides-T14087	7174	24	MgGI_library_#8MD	310
2	Coccidioides_T14926	5807	25	MgGI_library_#CN7	265
3	FvGI_library_T13047	5238	26	MgGI_library_#A39	206
4	Coccidioides_T14927	5213	27	MgGI_library_#8MC	165
5	Coccidioides_T14088	3894	28	BtGI_library_7121	154
6	AfGI_library_T10963	2949	29	MgGI_library_#1B9-mag	152
7	FvGI_library_T11287	2902	30	FvGI_library_T12441	52
8	FvGI_library_T11852	2880	31	FvGI_library_T12985	52
9	MgGI_library_#F3P	2864	32	FvGI_library_T12984	47
10	CrGI_library_CJ21	2678	33	MgGI_library_#F27	41
11	AnGI_library_#11R	2504	34	AfGI_library_T12734	31
12	FvGI_library_T10964	2319	35	AnGI_library_#1IE	29
13	AfGI_library_T11915	1929	36	ScGI_library_864-sachharo	28
14	FvGI_library_T13474	1907	37	FvGI_library_T14018	24
15	MgGI_library_#F3J	1780	38	AnGI_library_#0D3	14
16	MgGI_library_#CN6	1541	39	ScGI_library_106-saccharo	6
17	CrGI_library_CH99	1525	40	MgGI_library_#F0L	5
18	FvGI_library_T12440	1407	41	MgGI_library_#8M4	4
19	AnGI_library_#EP5	1198	42	ScGI_library_289-sacharo	3
20	SpGI_library_1380	909	43	BtGI_library_#5D0	2
21	MgGI_library_#BKH	768	44	CrGI_library_487	2
22	MgGI_library_#A36	716	45	AfGI_library_T13687	0
23	ScGI_library_241-sachhao	571	46	MgGI_library_#158-magna	07

string-matching algorithm known as *Knuth-Morris-Pratt* (KMP) algorithm [15] by a running time complexity of O $(n+m)$ i.e, in the worst case the algorithm has to examine all the characters in the text and pattern at least once.

This algorithm uses information about the characters in the pattern to determine how much to move along that string after a mismatch occurs. Pattern P is preprocessed by computing a failure function f that indicates the largest shift s using previously performed comparisons. Specifically, the failure function $f(j)$ is defined as the length of the longest prefix of P that is a suffix of $P[i..j]$. Note that the failure function f for P, which maps j to the length of the longest prefix of P that is a suffix of P[1.. j], encodes repeated substrings inside the pattern itself. The KMP algorithm is depicted in Fig. 3.

RESULTS

A database composed of fungi DNA sequences were gathered from different genomics engines [16-18] and searches of specific string with different alphabets long was conducted with the MAS structure proposed. Table 1 shows a sample query result in which user wants to find the most relevant micro organism class with respect to 4 subsequences of ATTATAGC, ACCATAGA, AGTCCAAC and ATTACGGC.

CONCLUSION

The problems faced by biological scientist in relation to increasing amounts of genomic data being generated are becoming critical. Through the use of MAS for pattern search in a sample database, it can be concluded that MAS is a beneficial tool for string searches in DNA pattern search and may be used in the analysis of sets of sequences from complete genomes. Finally, the next step will be construction of agents to link and analysis gene expression data using some soft-computing mechanisms.

REFERENCES

1. Andrade, M.A., N.P. Brown, C. Leory, S. Hoersch, C. Reich, A. Franchini, J. Tamames, A. Valencia, C. Ouzounis and C. Sander. 1999. Automated genome sequence analysis and annotation. Bioinformatics, 15: 391-412.
2. Hertz, G. and G. Stormo, 1995. Identification of consensus patterns in unaligned DNA and protein sequences: a large-deviation statistical basis for penalizing gaps. 3rd International Conference Bioinformatics and Genome Research, World Scientific, pp: 201-216.

3. Proutski, V. and E.C. Holme, 1996. Primer Master: A new program for design and analysis of PCR primers, CABIOS, 12: 253-255.
4. Stormo, G., Consensus patterns in DNA. In: Doolittle, D.F. (Ed.), Molecular evolution: Computer analysis of protein and nucleic acid sequences, Methods in Enzymol., 183: 211-221.
5. Ashburner, M., C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin and G. Sherlock, 2000. Gene Ontology: Tool for the unification of biology. Nat. Genet., 25: 25-29.
6. Baker, P.G., A. Brass, S. Bechhofer, C. Goble, N. Paton and R. Stevens, 1998. TAMBIS--Transparent Access to Multiple Bioinformatics Information Sources. Intl. Conf. Intell. Syst. Mol Biol.
7. Bechhofer, S., F. van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-chneider and L.A. Stein, 2004. OWL Web Ontology Language W3C Recommendation February 2004.
8. Christopher, J. and O. Baker, 2004. Greg Butler, Volker Haarslev, Ontologies, semantic web and intelligent systems for genomics, the 1st Canadian Semantic Web Interest Group Meeting (SWIG'04), November 19, 2004.
9. Fungal Genomics at Concordia: http://fungalgenomi cs.concordia.ca/home/index.php
10. Fabio Bellifemine, Giovanni Caire, Tiziana Trucco, Giovanni Rimassa, JADE Programmer's Guide, 21-August-2006.
11. JADE Board, JADE WSIG Add-On Guide JADE Web Services Integration Gateway (WSIG) Guide, 03-March-2005.
12. Cortese, E., F. Quarta, G. Vitaglione and P. Vrba, 2002. Scalability and Performance of the JADE Message Transport System. Analysis of Suitability for Holonic Manufacturing Systems, exp, 3: 52-65.
13. Shenghua, Liu, Peep Küngas and Mihhail Matskin, 2006. Agent-Based Web Service Composition with JADE and JXTA, Proceedings of the 2006 International Conference on Semantic Web and Web Services, SWWS 2006, Las Vegas, Nevada, USA, June 26-29, 2006.
14. Gradecki, J.D., 2002. Mastering JXTA: Building Java Peer-to-Peer Applications, John Wiley and Sons, 2002.
15. Cormen, T.H., C.E. Leiserson, R.L. Rivest and C. Stein, 2001. Introduction to Algorithms, 2nd Edn. (2001), MIT Press and McGraw-Hill, 923-931. ISBN 0262032937.
16. http://compbi.dfci.harvard.edu/tgi/fungi.html
17. http://www.123genomics.com
18. http://broad.mit.edu/annotion/fungi/genome.gov