Agent Technology in Bioinformatics: A Review

GURPREET SINGH BHAMRA
Maharishi Markandeshwar University at Mullana
ANIL KUMAR VERMA
Thapar University at Patiala
and
RAM BAHADUR PATEL
Gobind Ballabh Pant Engineering College at Ghurdauri

Mobile Agent is an autonomously transportable code performing node to node migration without losing its operability. Advantages of intelligent agent technology are being effectively utilized in simulating and modeling of complex biological systems. The main aim of this study is to motivate the researchers into the use of intelligent software agent technology in bioinformatics domain by providing an overview and updated comparison of the current agent enriched biological systems.

Keywords: Software Agents, Multi Agent Systems, Bioinformatics, Workflow Management System

1. INTRODUCTION

Bioinformatics or computational molecular biology aims at automated analysis and the management of high-throughput biological data as well as modeling and simulation of complex biological systems with the help of mathematical and computer science techniques. The first sequence alignment algorithms [Gibbs and McIntyre 1970] appeared in 1970s and since then Bioinformatics field has very much changed as it has now encompasses a wide range of subject areas from structural biology, genomics to gene expression studies. Today in-silico analysis is a fundamental component of biomedical research [Setubal and Meidanis 1997; Attwood and Parry-Smith 1999; Claverie 2000; Luscombe et al. 2001; Baxevanis and Ouellette 2001; Baldi and Brunak 2001; Lesk 2014; Ouzounis and Valencia 2003; Cohen 2004].

Intelligent software agent technology is an interdisciplinary technology inherited from Distributed Computing (DC), Distributed Artificial Intelligence (DAI), advanced knowledge base systems, and human computer interaction. The motivating idea of this technology is the development and efficient utilization of autonomous software objects called agents, which have access to geographically distributed and heterogeneous information resources to simplify the complexities of DC. They are autonomous, adaptive, reactive, pro-active, social, cooperative, collaborative and flexible. They also support temporal continuity and mobility (weak and strong) within the network. An intelligent agent with mobility feature is known as Mobile Agent (MA). MA migrates from node to node in a heterogeneous network without losing its operability. It can continue to function even if the user is disconnected from the network. It is carrying its code, execution state and other data while on the move. On reaching at a node MA is delivered to an agent execution environment where its authenticity is verified and executable parts are started running. Upon completion of the desired task, it delivers the results to the home node or to another node.

MAs can be viewed as an extension of the traditional client/server paradigm. With MA, a single serialized object is transmitted over the network carrying the small amount of resultant data only thus reducing the consumption of network bandwidth, latency(response time delay)

Authors' addresses: Gurpreet Singh Bhamra, Department of Computer Science & Engineering, M.M.University, Mullana-133203, Haryana, India; Anil Kumar Verma,Department of Computer Science & Engineering, Thapar University, Patiala-147004, Punjab, India; Ram Bahadur Patel, Department of Computer Science & Engineering, GBP Engineering College,Ghurdauri-246194, Uttarakhand, India.

and network traffic. They are robust , fault-tolerant and useful for low-cost, light weight, portable computing devices having the low processing powers, memory constraints, and intermittent low bandwidth connection. Agent's strong mobility feature is helpful in load balancing of processor and memory intensive tasks. A distributed application, built as a collection of MAs, can be easily extended with new functionality in the form of agents with slight modifications to the master agent. Number of participating hosts can be increased without any significant impact on the complexity of the application. MAs are self-contained and highly reusable. The parent agent can also clone several child agents to implement concurrent operations, and improve the efficiency. They also facilitate the rapid prototyping of distributed applications as software components can be flexibly and dynamically deployed in the form of MAs.

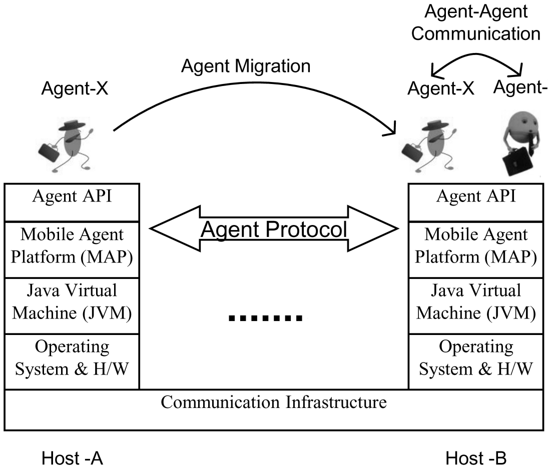
Agent-Agent Communication
Agent Migration
Agent-X
Agent-X
Agent-Y
Agent API
Mobile Agent Platform (MAP)
Java Virtual Machine (JVM)
Operating System & H/W
Agent Protocol
....
Agent API
Mobile Agent Platform (MAP)
Java Virtual Machine (JVM)
Operating System & H/W
Communication Infrastructure
Host -A
Host -B

Figure. 1: General architecture of a java based MAP [Bhamra et al. 2014]

A Mobile Agent Platform (MAP) a.k.a. Agent Execution Environment(AEE) or Agent Development (Tool)kit, is a middleware, distributed, server application that provides the appropriate functionality to MAs to authenticate, execute, communicate (with other agents, users, and other platforms), migrate to other platform, and use system resources in a secure way. A Multi Agent System (MAS) is distributed application comprised of multiple interacting intelligent agent components. The conceptual model of a java based MAP is depicted in figure 1. It consists of a) Host machine or a Network Node with Operating System and other hardware; b) Java Virtual Machine (JVM)-a runtime environment for platform independence and security; c) Mobile Agent Platform(MAP)- A multithreaded server application running on the top of JVM and provides the agent API with the basic functionality for agent management, migration, communication and security etc., in the form of java packages; and d) Communication Infrastructure (wired or wireless network) connecting the Host machines. Java MAs are special Java objects, executed as threads within an AEE. TCP/IP is used as the main transport mechanism. The behavior of a MAP is based on the services it offers. MAPs provide a set of core services and could be enriched with new services, implemented as service agents on top of the core functionality. The basic facilities provided by an AEE include Mobility, Communication, Security, Identification/Naming Scheme, Portability/Platform Heterogeneity, Resource Management, Resource Discovery, Control/Computation, Persistence, Interoperability, Scalability and Stability.

Today, agents are deployed in different settings and application domains such as Information Retrieval System(IRS); Systems and Network management; Mobile Computing; Managing Mails and Messaging; Workflow and Administrative Management; Electronic Commerce; Industrial control; Internet searching; Personal assistance; Games; Software distribution; Bioinformatics;

Clinical Data analysis for medical diagnosis; Distributed Data Mining(DDM); Intrusion Detection System(IDS); Peer-to-Peer Computing; Grid and Cluster Computing; Wireless Sensor Network(WSN); Unmanned Air and Under-Water Vehicles; Oil production systems etc [Bhamra et al. 2014].

2. AGENT TECHNOLOGY AND BIOINFORMATICS

A Workflow is the coordinated execution of multiple tasks or activities in a process applying a a set of procedural rules. A Workflow Management Systems (WMS) is a system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications [Hollingsworth 1994; WfMC 1999]. The biological experiment can also be seen as a workflow as bio-scientists performs their in-vitro experiments or in-silico daily work by executing a set of distinct, time consuming and repetitive activities in a distributed environment. In fact, they are always in need of flexible and expandable computational analysis and simulation tools[Steven et al. 2001; Merelli et al. 2002]. Academic research oriented and industrial WMSs have already been proposed and are being applied in the biomedical domain [Romano 2007].

Sequencing of organisms results into generation of huge quantity of dynamic, unstructured, heterogeneously distributed and exponentially growing raw biological data. Sharing this high-throughput biological data and making sense (or mining interesting knowledge) out of this data is a common problem among every one involved in this field [Bryson et al. 2000; Orro et al. 2005; Bartocci et al. 2006; Bartocci et al. 007a; Bartocci et al. 007b; Yang et al. 2009]. The lack of network bandwidth is also one of the main limitations on user activities which prohibits efficient transmission and receiving of huge biological data [Merelli et al. 2002]. The post-genomic era has resulted into availability of the enormous amount of distributed biological data sets that require suitable tools and methods for modeling and analyzing biological processes and sequences. The bioinformatics research community feels a strong need to develop new models and exploit and analyze the available genomes [Ezziane 2006].

The concept of data mining(DM) came to biology around 2000. Since then lots of DM algorithms, software tools, and web tools on bioinformatics have been proposed and designed. Software agents have been considered for DM from 2001. AgentLink (http://www.agentlink.org) funded by the European Commission's 6^{th} Framework Programme, was a Coordination Action for Agent Based Computing for providing support for researchers and developers in the domain of agent-based computing. The first Technical Forum was held from 30^{th} June to 2^{nd} of July, 2004 in Rome, Italy, at which the first BIOAGENTS Technical Forum Group founded the Working Group. Agent technology provides an appropriate solution for applications that deals with the repetitive,time-consuming,interactive and coordinated activities;learning, planning and information and knowledge management and modelling and simulations of complex, dynamic systems [Corradini and Merelli 2005; Shunmuganathan et al. 2008]. The advantages of software agents discussed above make the agent technologies and MASs suitable for bioinformatics field and constitute an active and emerging area both in relation to designing domain-aware MASs for simulating and modeling biological systems through autonomous components interactions; and for the automation of data collection and service discovery processes along with knowledge management and problem-solving. Interactions among agents in a MAS resembles to the interactions among entities in the modeled biological system. The agent abstraction allows us to directly model both reactive and proactive behavior of biological components and their environment [Luck et al. 2003; Luck et al. 2005; Luck and Merelli 2005; Orro et al. 2005; Merelli and Young 2005; Merelli et al. 2006]. The genome scientists can design reagents for future research in livestock genomics with the help of communities of software agents. Software agents are being used to assist in the process of submitting sequence data to GenBank at the Meat Animal Research Center (MARC) at Clay Center, Nebraska [Keele and Wray 2005]. Agent technology would be an increasingly important domain for biological data mining and analysis in coming years [Yang et al. 2009].

3. MAP/MAS IN BIOINFORMATICS

Use of intelligent software agent technology in bioinformatics started around 2000. Most of the research in bioinformatics using agent technology focuses on designing MAS for WFM [Bartocci2006]. Some prominent MAS in bioinformatics domain are GeneWeaver[Bryson et al. 2000], BioMAS[Decker et al. 2001; Decker et al. 2002], BioAgent[Merelli et al. 2002], CellMAS[Corradini et al. 2005; Merelli and Young 2005], Agent based framework for PSP(AbFPSP)[Bortolussi et al. 2005], BIOPACMAS[Orro et al. 2005], MAS for gene expression data (MAS_GED)[Lam et al. 2006], Multiagent based bio-data mining(MABDM)[Yang et al. 2009], MAS_LIMS[Maccagnan et al. 2010]. Description of these and some other systems are given below.

GeneWeaver [Bryson et al. 2000] is a prototype MAS designed for genome analysis and protein structure prediction. It composed of a community of task oriented agents. Each agent encapsulates the existing databases and tools and interact with other agents to effectively automate the biological processes. Various agents involved in *GeneWeaver* agent community are:-

- —PDB ,Swiss and PIR Agents - To manage the primary sequence databases.
- —Protein NRDB (non-redundant database) Agent - To integrate the data managed by PDB Agent, Swiss Agent and PIR Agent.
- —Calculation, Blast and Clustal Agents - To annotate biological sequences using relevant programs. The skills of the various calculation agents can be combined by an expert calculation agent using expert knowledge encoded in plans. In this case it can use the Blast Agent to find similar proteins and then use the Clustal Agent to accurately compare the proteins obtained.
- —Genome Agent - To provide functionality of external access to the results generated by the system via the WWW.

Each agent in the *GeneWeaver* community shares a common AEE comprising the generic modules of control, motivation, action, interaction, and communications. The motivation module gives direction to a central control module through particular goals in a best possible way. It can either take action itself through its action module, or request assistance from another agent through its interaction module and the interaction mechanism is handled by the communications module. The meta-store acts as a repository for local information such as the skills of other agents. Neither the data store nor the analysis tools are regarded as a part of the agent but they are used by agents either to store persistent data they are working with or to analyze the data.

BioMAS [Decker et al. 2001; Decker et al. 2002] is a prototype MAS based on *DECAF* [Graham et al. 2003] toolkit for automated annotation and database storage of sequencing data for herpesviruses. The resulting system eliminates tedious and hand analyses, makes the data and annotations available for other researchers, and provides a level of query processing beyond even some high-profile web sites. *BioMAS* uses the distributed and open nature of its multi-agent solution to expand the information gathering system in several ways that will make it useful for biologists studying more organisms, and in different ways. Based upon the *BioMAS*, Jin et al. [2005] present a new approach to visualize complex datasets representing gene expression and pathway models in hierarchical view space using the Starlight information visualization system.

BioAgent [Merelli et al. 2002]is a MAS to support bio-scientists during the process of genome analysis and annotation. The 4-layered software architecture of *BioAgent* system consists of the Core layer, Service agents layer, Bio agents layer, and Workflow layer. The Core layer provides basic features of any MAP; the Service agents layer consists of the set of services available in a single place; the Bio agents layer hosts and manages the set of mobile user agents. And the Workflow layer provides a definition language, called BioAgent-l, is appropriate and provides ease of handing for the definition of a bio-scientist workflow in terms of coordinated execution of multiple tasks or activities.

Hermes [Bonura et al. 2003; Corradini et al. 2003; 2004; Corradini and Merelli 2005]is a 3-layred, component-based, AEE completely developed in Java for design and execution of activity or workflow based applications in distributed environments. It is flexible, modular, scalable, and can be easily configured for specific application domains by adding domain-specific component libraries. Application domain experts can focus on designing activity workflow and ignore the topological structure of the distributed environment. Figure 2 shows the layout for a distributed system designed over *Hermes* middleware in heterogeneous environment.

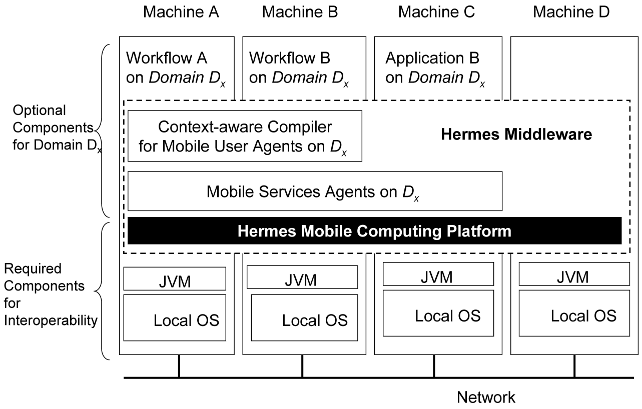
Machine A
Machine B
Machine C
Machine D
Workflow A on *Domain D_x*
Workflow B on *Domain D_x*
Application B on *Domain D_x*
Optional Components for Domain D_x
Context-aware Compiler for Mobile User Agents on D_x
Hermes Middleware
Mobile Services Agents on D_x
Hermes Mobile Computing Platform
Required Components for Interoperability
JVM
Local OS
JVM
Local OS
JVM
Local OS
JVM
Local OS
Network

Figure. 2: A Distributed System over Hermes Middleware [Corradini and Merelli 2005]

The 3-layred software architecture of *Hermes* consists of the user layer, system layer and run-time layer. The core abstract Agent class assigns the basic features to each agent such as mobility with two associated extensions UserAgent and ServiceAgent. At the user layer, it allows designers to specify their applications as a workflow of activities using the graphical notation provided by Enhydra JaWE workflow editor [JAWE 2003]. The workflow editor is used to compose the primitive as well as complex activities. A primitive activity can be directly executed. A complex activity must be specified before it can be used and such specification further is a workflow of complex and simple activities. , A context-aware compiler at the system layer generates a pool of user mobile agents from the workflow specification. The design phase of the compiler relies in a two steps: 1) the User-Level Activity Database (ULAD) is used to map the User Level Workflow (ULW) to Agent Level Workflow (ALW); 2) the User-level Activity Implementation Database (ULAID) and the Database of Skeletons (DoS) are used to code the ALW in a pool of mobile Workflow Executors(WEs) , i.e., the UserAgents. At the run-time layer, it supports the activation of a set of specialized ServiceAgents, and it provides all necessary to support agent mobility,security, fault tolerance, communication, and resource management and discovery. ServiceAgents act as guard of the local resources and are created any time a new resource becomes available in the distributed environment. UserAgents (WEs) interact with distributed environment through ServiceAgents (SAs). In fact, when a UserAgent migrates and arrives in a different platform, it can query the YellowPageService to gain information about services offered in the platform and then it communicates with ServiceAgents to gain the information it needs. Further these 3 layers are themselves split in two conceptual levels: - the type of application running on each layer and - the infrastructure supporting the application. At the user layer, the application is the workflow and the infrastructure is the workflow management environment (editor, model checker etc.). At the system layer, the application is a pool of running UserAgents or WEs, and the infrastructure is given by the agent management system (compiler, model checker, query optimizer, . . .). Finally, at the run-time layer, the application is given by a set of services provided by ServiceAgents and the infrastructure consists of the mobile computing platform for agents mobility. Main Services and Tools for Hermes include, XML Generalized Wrapper, i.e., Any Input XML Output(AIXO) Service Agent; Web Service Service Agent(WS2A); Matchmaker Service Agent(MSA); Ontology Management Service Agent(OMSA); light Tuple Space Service agent(lightTS-SA); Web Interface Service Agent(WISA); and Workflow Interface Service Agent(WfSA). Details of these services can be found in [Corradini and Merelli 2005].

CellMAS [Corradini et al. 2005; Merelli and Young 2005] is a MAS designed on the top of *Hermes* for modeling Carbohydrate Oxidation in cellular process. A biological cell is a complex system comprising of a large number of interacting components within a dynamic environment. Each cell acts as a self-contained and autonomous entity, i.e., it can take in nutrients, convert these nutrients into energy, and reproduce itself. Metabolic pathways are the series of chemical reactions within a cell that realise cell metabolism. Carbohydrate oxidation is a set of metabolic pathways by which a cell produces energy through chemical transformation of carbohydrates like fructose, glucose, etc. The *CellMAS* models the structure as well as the behavior of the biological system, the metabolic pathways are encapsulated and distributed among software agents that model the cellular components that collaborate in each pathway.

BioWMS [Bartocci et al. 2006; Bartocci et al. 007a; Bartocci et al. 007b] is another WMS case study for the use of *Hermes V2* middleware in Open Grid Services Architecture (OGSA)[Foster et al. 2003] compatible Grid computing environment. In this case study authors used Petri nets-based notations for the compilation phase at the system layer of *Hermes* middleware. In a two steps compilation phase, User Workflow Specifiations (UWSs) are mapped onto agent level Interactive Components (ICs) comprising a MAS. In the first step, UWS is translated to a Role-based Workflow Specification (RWS) The first translation assigns the resources or roles needed to execute each task. Wf-net [van der Aalst 1998], a well-known extension of classical Petri net [Murata 1989], is chosen as high-level specification language suitable to represent the main workflow patterns. The second step translates RWS into Interactive Components and in order to specify the behavior of each component independently from the corresponding generated code, BRICs [Ferber 1999], another Petri nets-based notation, is used to translate Wf-net to BRICs notation. *BioWMS* provides a Web-based interface to define in-silico experiments as workflows of complex and primitive activities. It provides a web-based editor, called WebWFlow, that enables the definition of a workflow in the XML Process Definition Language (XPDL). Workflow client applications, like BioWEP [Romano et al. 2007], are used to execute existing or previous saved workflows, to check the workflow execution state and to manage the produced results. The workflow enactment service provides the run-time environment in which process instantiation and activation occur, utilising one or more workflow management engines, responsible for interpreting and activating part, or all, of the process definition and interacting with the external resources necessary to process the various activities. The workflow enactment in *BioWMS* is supported by the *Hermes* middleware and its context aware compiler (XpdlCompiler). The *Hermes* Graphical User Interface (GUI) is an administration and monitoring tool that allows checking, at any time, of the computational resources available, the memory consumption and the agents that are running.

In the OSGA compatible multilayer Grid setup, in-silico experiments can be organized as workflows to be executed on *BioWMS*. Agent-based simulation of biological process takes place in proposed Orion component. In its present version it takes into account behavioral modeling of metabolic reactions. Resourceome is an ontological model that keeps an alive semantic index of resources in the bioinformatics domain. Simultaneously, *BioWMS* and Orion can interact with Resourceome, taking advantage of the same machine-understandable knowledge infrastructure. An OGSA infrastructure allows a transparent and controlled access to the Grid resources, both in the biological systems simulations and in-silico experiments. The interactions among OGSA, *BioWMS*, Resourceome, and Orion are agent-driven. This proposed virtual laboratory is designed for the Laboratory for Interdisciplinary Technologies in Bioinformatics (LITBIO), a FIRB project applied to Genomics and Proteomics research at an international level [Corradini 2006].

Karasavvas et al. [2004] proposed the use of agent technology for designing bioinformatics integration systems. Bortolussi et al. [2005] presented agent based framework for protein structure prediction (PSP) problem of predicting the 3D native conformation of a protein from the known sequence of amino acids, implemented using the Linda package of SICStus Prolog and C++. Each amino acid is represented by independent software agent that communicates with the others. Delgado et al. [2005] presented the development of *BioMen* (Biological Management Executed over Network) managed by means of a MAS.

BIOPACMAS[Orro et al. 2005] is a Personalized, Adaptive, and Cooperative Multi Agent System (PAC-MAS) designed to predict protein secondary structure. *PACMAS* agents are autonomous and flexible, and can be personalized, adaptive, and cooperative-depending on the implemented application. It is a generic multi agent architecture designed to support the implementation of applications aimed at: (i) retrieving heterogeneous data spread among different Internet sources, (ii) Filtering and organizing them according to personal interests explicitly stated by each user, and (iii) providing adaptation techniques to improve and refine throughout time the profile of each selected user. The overall architecture consists of four main levels: information, filter, task, and interface. Each level is associated to a specific role and populated by a society of agents, so that communication may occur both horizontally and vertically. The information agents are able to access information sources to collect and manipulate such information. Agents play the role of wrappers at this level. Task agents are able to help users in various problem-solving activities and in exchanging information with other agents. Interface agents are in charge of interacting with the user other agents.The communication between adjacent levels is achieved through suitable middle agents, which form a corresponding mid-span level.

Lam et al. [2006] proposed a multi-agent approach to conduct the analysis of gene expression data generated using microarray technology. The system supports concurrent gene data processing by implementing independent agent solutions and interface agent solutions to coordinate data flow. The MAS is built on top of the Bioconductor which is an Open Source Software for Bioinformatics (www.bioconductor.org), a platform in which a wide range of statistical and graphical methods for the analysis of high-throughput genomic data can be incorporated. The Multi-agent system architecture is composed of various interface agents and individual working agents. Master Agent (MA) is the core of the system that is responsible for distributing work to different agents, coordinate activities among agents, interacts with interface agents by issuing Master Command Sequence (MCS), and store agents information in its agent repository. In order to maintain control of all the agents in the system, the master agent also implements an agent repository database which keeps information of all the available agents such as Agent Name(unique), Agent Service, Agent Status, and Agent Date. A new working agent can register with the master agent by providing its name and the service it implements. The working agents are dynamic system components, which means that they can either be registered with or deregistered with the master agent. Various agents and their responsibilities are described as below: Once the microarray experiment is finished and the necessary data are acquired, the system starts with a user interacting with the Query Agent to input the parameter setting for the analysis. Once the required information has been gathered, the query agent arranges the sequence of options chosen by the user as a colon delimited User Input Sequence (UIS) and forwards it to the UIA., which in turn forwards it to the MA. The MA will then decide what data processing path the gene data will travel in the system. Once the data processing path has been determined, the MA will hand the data to the Preprocessing Interface Agent (PIA) which in turn will send the data to a Normalization agent (N) to perform data normalization if specifically requested. After normalization, the PIA collects the data output from the normalization agent and sends it to the Statistical Analysis agent (S) through the Analysis Interface Agent (AIA). The statistical analysis agent will perform requested statistical analysis with the data and output a gene list with genes that are found to have statistical significant expression values. This gene list can then be sent to DB interface agent (DBIA). The DBIA then invokes the proper DB agent for functional analysis. Once the result become available, DBIA passes the result back to the MA and it then will forward it to the UIA. The UIA sends the result to the Result Agent which will format and output it to the user. The user can then make the final verification and conclusion based on the result derived by the system.

Yang et al. [2009] proposed a prototype MAS implemented using JADE for biological data analysis. The technical details of choosing mining algorithms, forming hybrid system, and preparing specific data format relies upon the intelligent agent-based bio-data mining system itself. In this 3-level framework(see figure 3), the first level comprises of the interface agents to collect mining task from end users and also search for available planning and aggregation agent from the yellow page. The second level comprises of the planning agent for task planning and aggregation agent for mining results aggregation. The third level are the mining agents. Each mining agent manages a geographically distributed database using the mining knowledge and a pack of mining algorithms. Each mining agent's service is registered in the yellow page. The interface agents search for planning agent at the yellow page, record its address and sends the mining task collected from the end user to the available planning agent. The planning agent then searches for mining agents which is capable to provide mining results with respect to this task. When the candidate mining agent(s) has been identified, the task or its subtask is deployed to the mining agent(s) and the aggregation agent is informed. When the mining results are available from any mining agent, it sends the results back to the aggregation agent after identifying its address in the yellow page. The aggregation agent combines all the mining results and sends them back to the interface agent for display. A bioinformatics ontology base is used to match the task with multiple data sources.

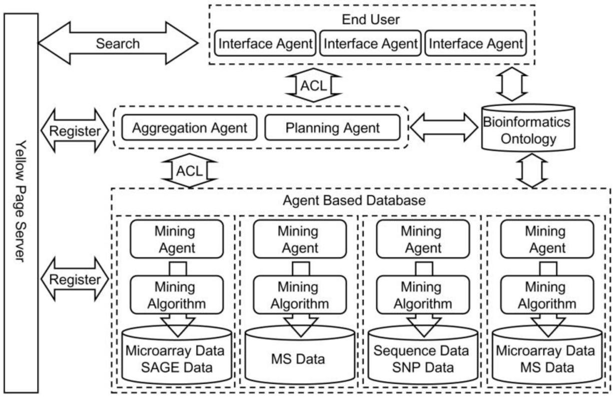

Figure. 3: Multiagent based bio-data mining framework [Yang et al. 2009]

Maccagnan et al. [2010] designed a MAS component of a Next Generation Laboratory Information Management Systems (LIMS) project built upon the meta-model called Combining Ontologies and Workflows (COW). Next generation LIMS project is combines Model-Driven Engineering, Workflows, Ontologies and MAS. MAS has been implemented with JADE and WADE platforms to automatically interpret and execute a structured representation of laboratory protocols (precisely defined sequence of activities for laboratory experiments and operations) expressed in XML Process Definition Language [XPDL 2012]and Web Ontology Language [OWL 2004]. Various agents involved are Device Agent(DA)-to control heterogeneous resources, Protocol Manager agent (PM)-to execute a protocol, Agent Protocol Environment (APE) agent - acts as user interface agent to allow the loading of new protocols , Reporter Agent(RA)- acts as user interface for mobile devices.

Bazzan [2009]discussed the use of DM and MAS for genome annotation pipeline process and Gao et al. [2011] proposed a MAS for remote data retrieval for bioinformatics applications and discussed the synchronous and asynchronous migration strategies for agents.Recently Chaouch and Tamali [2014] proposed an Android based MAS (DiabMAS) for remote medical monitoring of diabetic patients on an outpatient. The main aim is for the improvement of the transmission of information between patients and their physicians, especially the management of specific and critical cases. Qualitative comparison of some prominent current MAS in bioinformatics domain is provided in Table I taking into account some of the features they provide. The features include the following fields:-

—**BD** indicates which biological domain MAS addresses.

—**WMS** indicates whether MAS acts as a Workflow Management Systems.

—**Agents** shows the community of agents involved in the system.

—**AEE/MAP** shows which underlying Agent Executing Environment or Mobile Agent Platform is used for developing MAS.

—**Impl** field indicates whether the MAS is implemented along with the language used for implementation or it is just a Prototype model.

—**FW** indicates the further work required in the system under study.

Table I: Qualitative comparison of Multi Agent Systems in Bioinformatics

MAS	**Features**	**Description**
GeneWeaver	BD	Genome analysis and protein structure prediction
	WMS	No
	Agents	PDB, Swiss, PIR, Protein NRDB, Calculation, Blast, Cluster and Genome agents
	AEE/MAP	A generic AEE is proposed
	Impl	No (Prototype only)
	FW	Implementation and validation of MAS and underlying AEE is required
BioMAS	BD	Genome annotation and sequencing for herpesviruses
	WMS	No
	Agents	Information extraction, Annotation, Sequence Source processing, Proxy, Query Processing, Matchmaker and Agent Name server agents.
	AEE/MAP	DECAF
	Impl	No (Prototype only) as `http://udgenome.ags.udel.edu/herpes/`link not operative
	FW	Implementation and ontology support for MAS is required.
BioAgent	BD	Genome analysis and annotation
	WMS	Yes
	Agents	User agent, Bio agents, Service agents for web interface, wrappers and ontology services and Broker agent
	AEE/MAP	Core Layer provides the features of an AEE
	Impl	Yes (Java, BioAgent-l)
	FW	System is constantly evolving by adding more functionality and has to prove itself in a complex real application
CellMAS	BD	Modeling the Carbohydrate Oxidation in cellular process
	WMS	No
	Agents	Agents encapsulating metabolic pathways
	AEE/MAP	Hermes
	Impl	Simulation model
	FW	Verification and validation required
AbFPSP	BD	Protein structure prediction
	WMS	No
	Agents	Each agent represents an Amino acid
	AEE/MAP	No
	Impl	Yes (Linda package of SICStus Prolog and C++)
	FW	AEE is required for this simulation engine

MAS	Features	Description
BIOPACMAS	BD	Protein structure prediction
	WMS	No
	Agents	Information, Filter, Task, Interface and Mid-span agents
	AEE/MAP	No
	Impl	No (Prototype only)
	FW	Implementation, verification and validation required. Underlying AEE required
MAS_GED	BD	Analysis of gene expression data
	WMS	No
	Agents	Statistical, Database, Analyze, Query, Result, Normalization and Interface agents
	AEE/MAP	Bioconductor
	Impl	No (Prototype only)
	FW	Verification and validation required. More machine learning algorithms required
MABDM	BD	Biological data mining
	WMS	No
	Agents	Interface, Aggregation, Planning and Mining agents
	AEE/MAP	JADE
	Impl	No (Prototype only)
	FW	Implementation, verification and validation required
MAS_LIMS	BD	Laboratory Information Management Systems to automatically interpret and execute a structured representation of laboratory protocols
	WMS	Yes
	Agents	Device, Protocol Manager, Agent Protocol Environment, Reporter Agents
	AEE/MAP	JADE and WADE
	Impl	Yes(Java, XPDL,OWL)
	FW	N/A

This study shows that most of the MAS in bioinformatics domain are prototype based academic research projects designed for a specific biological problem with no implementation, validation and verification and none of these is a commercial product. Hermes is the only AEE acting as middleware specifically designed for WMS (like BioWMS) for biological domain in a heterogeneous environment. Genome analysis and annotation, Protein structure prediction, modeling and simulation of biological process, workflow management system are the main areas in bioinformatics where the software agent technology is being currently used. DECAF, JADE and WADE are the prominent AEE used for designing a MAS. Java, Perl, Linda package of SICStus Prolog and C++ are the main programming languages in use for implementing the MAS. Lots of work is still required for actually implementing such MAS in a distributed environment using real biological data sets; applying classification, clustering and association rules mining techniques within the real biological datasets.

4. CONCLUSION

The extensive and growing resources of distributed biological databases and analysis tools is the reason behind the bioinformatics providing an appropriate domain for the application of agent technologies. It offers the possibility for deploying and testing agent systems in a real-world setting with the possibility of making substantial contributions to human society. On the other hand, there is a distinct and identified need for good solutions to improve the performance of existing bioinformatics systems, and agents may be able to contribute to that improvement. In this sense, there is a very strong synergy between the two domains. However, there are still significant challenges. Researchers from both communities generally require education in each other's domain, and there must be mutually acceptable solutions across both areas. Instead of designing prototype MAS, focus should be on actual implementation using the real biological dataset. With this study, we expect to contribute to cover the need of an updated review of the role of intelligent agents in bioinformatics and also to encourage future work in this domain.

REFERENCES

Attwood, T. and Parry-Smith, D. 1999. *Introduction to Bioinformatics*, First ed. Prentice Hall. ISBN:0582327881.

Baldi, P. and Brunak, S. 2001. *Bioinformatics: The Machine Learning Approach.* MIT Press. ISBN:0-262-02506-X.

Bartocci, E., Cacciagrano, D., Cannata, N., Corradini, F., Merelli, E., Milanesi, L., and Romano, P. 2007a. An agent-based multilayer architecture for bioinformatics grids. *IEEE Transactions on NanoBioscience 6,* 2, 142–148.

Bartocci, E., Corradini, F., and Merelli, E. 2006. Building a MultiAgent System from a User Workfow Specifcation. In *Proceedings of the 7th WOA 2006 Workshop, From Objects to Agents (Dagli Oggetti Agli Agenti)*, F. De Paoli, A. Di Stefano, A. Omicini, and C. Santoro, Eds. Catania, Italy, 96–103.

Bartocci, E., Corradini, F., Merelli, E., and Scortichini, L. 2007b. BioWMS : a web-based Workflow Management System for bioinformatics. *BMC Bioinformatics 14*, 1–14.

Baxevanis, A. and Ouellette, B. 2001. *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*, Second ed. Willey & Sons. ISBN:0-471-38391-0.

Bazzan, A. L. 2009. Agents and Data Mining in Bioinformatics: Joining Data Gathering and Automatic Annotation with Classification and Distributed Clustering. In *Agents and Data Mining Interaction*, L. Cao, V. Gorodetsky, J. Liu, G. Weiss, and P. S. Yu, Eds. Lecture Notes in Computer Science, vol. 5680. Springer Berlin - Heidelberg, 3–20. See also HermesV2 http://hermes.cs.unicam.it.

Bhamra, G. S., Patel, R. B., and Verma, A. K. 2014. Intelligent Software Agent Technology: An Overview. *International Journal of Computer Applications(IJCA) 89,* 2 (March), 19–31.

Bonura, D., Mariani, L., and Merelli, E. 2003. Designing modular agent systems. In *Proceedings of NET.Object DAYS.* Erfurt, 245–263.

Bortolussi, L., Dovier, A., and Fogolari, F. 2005. Multi-Agent Simulation of Protein Folding. In *Proceedings of the 1st AAMAS International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics (MAS*BIOMED'05).* Utrecht,Netherlands, 91–106.

Bryson, K., Luck, M., Joy, M., and Jones, D. 2000. Applying agents to bioinformatics in Geneweaver. In *Cooperative Information Agents IV- The Future of Information Agents in Cyberspace*, M. Klusch and L. Kerschberg, Eds. Lecture Notes in Artificial Intelligence, vol. 1860. Springer-Verlag, 60–71.

Chaouch, Z. and Tamali, M. 2014. A Mobile Agent-Based Technique for Medical Monitoring (Supports of Patients with Diabetes). *International Journal of Computational Models and Algorithms in Medicine (IJCMAM) 4,* 1 (March), 17–32.

Claverie, J. 2000. From Bioinformatics to Computational Biology. *Genome Research 10,* 9, 1277–1279.

Cohen, J. 2004. Bioinformatics - An Introduction for Computer Scientists. *ACM Computing Surveys 36,* 2 (June), 122–158.

Corradini, F. 2006. Laboratory for Interdisciplinary Technologies in Bioinformatics (LITBIO).

Corradini, F., Mariani, L., and Merelli, E. 2003. A programming environment for global activity-based applications. In *Proceedings of the 4th AI*IA/TABOO Joint Workshop "From Objects to Agents": Intelligent Systems and Pervasive Computing*, G. Armano, F. De Paoli, A. Omicini, and E. Vargiu, Eds. 163–169.

Corradini, F., Mariani, L., and Merelli, E. 2004. An agent-based approach to tool integration. *International Journal on Software Tools for Technology Transfer 6,* 3 (August), 231–244.

Corradini, F. and Merelli, E. 2005. Hermes: Agent-Based Middleware for Mobile Computing, Formal Methods for Mobile Computing. In *Formal Methods for Mobile Computing*, M. Bernardo and A. Bogliolo, Eds. Lecture Notes in Computer Science, vol. 3465. Springer-Verlag, 234–270.

Corradini, F., Merelli, E., and Vita, M. 2005. A multi-agent system for modelling carbohydrate oxidation in cell. In *Computational Science and Its Applications - ICCSA 2005*, O. Gervasi, M. L. Gavrilova, V. Kumar, A. Lagan, H. P. Lee, Y. Mun, D. Taniar, and C. J. K. Tan, Eds. Lecture Notes in Computer Science, vol. 3481. Springer Berlin - Heidelberg, 1264–1273.

Decker, K., Khan, S., Schmidt, C., Situ, G., Makkena, R., and Michaud, D. 2002. Biomas: A multi-agent system for genomic annotation. *International Journal of Cooperative Information Systems 11,* 3-4, 265–292.

Decker, K., Zheng, X., and Schmidt, C. 2001. A Multi-Agent System for Automated Genomic Annotation. In *Proceedings of the 5th International Conference on Autonomous Agents(AGENTS01).* ACM, Montreal, Quebec, Canada, 433–440.

Delgado, M., Fajardo, W., Gibaja, E., and Perez-Perez, R. 2005. BioMen: An information system to herbarium. *Expert Systems with Applications 28,* 3, 507–518.

Ezziane, Z. 2006. Applications of artificial intelligence in bioinformatics : A review. *Expert Systems with Applications 30*, 2–10.

Ferber, J. 1999. *Multi-agent systems: an introduction to distributed artificial intelligence.* Vol. 1. Addison-Wesley Reading.

Foster, I., Kesselman, C., Nick, J. M., and Tuecke, S. 2003. The physiology of the grid. In *Grid computing: making the global infrastructure a reality*, F. Berman, G. C. Fox, and A. J. G. Hey, Eds. Wiley Series on Parallel and Distributed Computing. John Wiley & Sons, 217–249. ISBN:0-470-85319-0.

Gao, L., Dai, H., Zhang, T.-L., and Chou, K.-C. 2011. Remote Data Retrieval for Bioinformatics Applications: An Agent Migration Approach. *PLoS ONE 6,* 6, 1–7.

Gibbs, A. and McIntyre, G. 1970. The diagram, a method for comparing sequences. *European Journal of Biochemistry 16*, 1–11.

Graham, J. R., Decker, K., and Mersic, M. 2003. DECAF - A Flexible Multi Agent System Architecture. *Autonomous Agents Multi-Agent Systems 7,* 1-2, 7–27.

Hollingsworth, D. 1994. Workflow Management Coalition: The Workflow Reference Model. Tech. Rep. WFMC-TC-1003 Draft 1.1, Workflow Management Coalition(WfMC). November.

JAWE. 2003. Together WfMC XPDL and OMG BPMN Workflow Editor.

Jin, L., Steiner, K. V., Schmidt, C. J., Situ, G., Kamboj, S., Hlaing, K. T., Conner, M., Kim, H., Emara, M., and Decker, K. S. 2005. A Multiagent Framework to Integrate and Visualize Gene Expression Information. In *IEEE ICDM Workshop on MADW and MADM.* 1–7.

Karasavvas, K. A., Baldock, R., and Burger, A. 2004. Bioinformatics integration and agent technology. *Journal of Biomedical Informatics 37,* 3 (June), 205–219.

Keele, J. and Wray, J. 2005. Software agents in molecular computational bioloy. *Briefing in Bioinformatics 6,* 4, 370–379.

Lam, H., Garcia, M., Juneja, B., Fahrenkrug, S., and Boley, D. 2006. A Multi-agent Approach to Gene Expression Analysis. In *Proceedings of the 2^{nd} International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics (MAS*BIOMED'06).* Future University, Hakodate, Japan, 60–73.

Lesk, A. M. 2014. *Introduction to Bioinformatics*, Fourth ed. Oxford University Press. ISBN:978-0-19-965156-6.

Luck, M., McBurney, P., and Preist, C. 2003. Agent Technology: Enabling Next Generation Computing (A Roadmap for Agent-Based Computing). Tech. Rep. AgentLink II Version 1.0, The AgentLink Community. January. ISBN: 0854 327886.

Luck, M., McBurney, P., Shehory, O., and Willmott, S. 2005. Agent Technology:Computing as Interaction (A Roadmap for Agent Based Computing). Tech. Rep. AgentLink III, The AgentLink Community. September. ISBN: 085432 845 9.

Luck, M. and Merelli, E. 2005. Agents in bioinformatics. *Knowledge Engineering Review 20,* 2, 117–125.

Luscombe, N., Greenbaum, D., and Gerstein, M. 2001. What is bioinformatics? A proposed definition and overview of the field. *Methods of Information in Medicine 40,* 4, 346–358.

Maccagnan, A., Vardanega, T., Feltriny, E., Valley, G., Rivaz, M., and Cannata, N. 2010. Multi-Agent System for the automated handling of experimental protocols in biological laboratories. In *Proceedings of the 11^{th} Workshop on Objects and Agents(WOA 2010),Dagli Oggetti Agli Agenti*, A. Omicini and M. Viroli, Eds. Rimini, Italy.

Merelli, E., Armano, G., Cannata, N., Corradini, F., dInverno, M., Doms, A., Lord, P., Martin, A., Milanesi, L., Mller, S., Schroeder, M., and Luck, M. 2006. Agents in Bioinformatics, Computational and Systems Biology. *Briefings in Bioinformatics 8,* 1 (May), 45–59.

Merelli, E., Culmone, R., and Mariani, L. 2002. Bioagent: A mobile agent system for bioscientists. In *Network Tools and Applications in Biology(NETTAB) Workshop - Agents in Bioinformatics.*

Merelli, E. and Young, M. 2005. Validating MAS Models with Mutation. In *Proceedings of the 1^{st} AAMAS International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics (MAS*BIOMED'05).* Utrecht,Netherlands, 146–156.

Murata, T. 1989. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE 77,* 4, 541–580.

Orro, A., Saba, M., Vargiu, E., and Mancosu, G. 2005. Using a Personalized, Adaptive and Cooperative MultiAgent System to Predict Protein Secondary Structure. In *Proceedings of the 1^{st} AAMAS International Workshop on Multi-Agent Systems for Medicine, Computational Biology, and Bioinformatics (MAS*BIOMED'05).* Utrecht,Netherlands, 170–183.

Ouzounis, C. and Valencia, A. 2003. Early bioinformatics: the birth of a discipline - A personal view. *Bioinformatics 19,* 17, 2176–2190.

OWL. 2004. OWL Web Ontology Language Reference. Tech. rep., World Wide Web Consortium (W3C). February.

Romano, P. 2007. Automation of in-silico data analysis processes through workflow management systems. *Briefings in Bioinformatics 9,* 1 (December), 57–68.

Romano, P., Bartocci, E., Bertolini, G., de Paoli, F., Marra, D., Mauri, G., Merelli, E., and Milanesi, L. 2007. BioWEP: a workflow enactment portal for bioinformatics applications. *BMC Bioinformatics 8,* Suppl 1 (March 8), S19.

Setubal, J. C. and Meidanis, J. 1997. *Introduction to Computational Molecular Biology.* PWS Publishing Company.

SHUNMUGANATHAN, K. L., DEEBA, K. V., AND DEEPIKA, K. 2008. Agent Based Bioinformatics integration Using RETSINA. *International Arab Journal of Information Technology 5,* 3, 258–264.

STEVEN, R., GOBLE, C., KAKER, P., AND A., B. 2001. A classification of tasks in bioinformatics. *Bioinformatics 17,* 2, 180–188.

VAN DER AALST, W. M. P. 1998. The application of Petri nets to workflow management. *Journal of circuits, systems, and computers 8,* 1, 21–66.

WFMC. 1999. Workflow Management Coalition: Terminology & Glossary. Tech. Rep. WFMC-TC-1011 Issue 3.0, Workflow Management Coalition(WfMC). Feb.

XPDL. 2012. Process Definition Interface - XML Process Definition Language(XPDL). Tech. Rep. WFMC-TC-1025 Version 2.2, Workflow Management Coalition(WfMC). August.

YANG, P., TAO, L., XU, L., AND ZHANG, Z. 2009. Multiagent Framework for Bio-data Mining. In *Rough Sets and Knowledge Technology*, P. Wen, Y. Li, L. Polkowski, Y. Yao, S. Tsumoto, and G. Wang, Eds. Lecture Notes in Computer Science, vol. 5589. Springer Berlin - Heidelberg, 200–207.

Gurpreet Singh Bhamra is currently working as Assistant Professor at Department of Computer Science and Engineering, M. M. University, Mullana, Haryana. He received his B.Sc. (Computer Sc.) and MCA from Kurukshetra University Kurukshetra in 1995 and 1998, respectively. He is pursuing Ph.D. from Department of Computer Science and Engineering, Thapar University, Patiala, Punjab. He is in teaching since 1998. He has published 8 research papers in International/National Journals and International Conferences. He has received Best Paper Award for "An Agent enriched Distributed Data Mining on Heterogeneous Networks", in "Challenges & Opportunities in Information Technology" (COIT-2008). He is a Life Member of Computer Society of India. His research interests are in Distributed Computing, Distributed Data Mining, Mobile Agents, Bioinformatics.

Dr. Anil Kumar Verma is currently working as Associate Professor at Department of Computer Science & Engineering, Thapar University, Patiala. He received his B.S., M.S. and Ph.D. in 1991, 2001 and 2008 respectively, majoring in Computer science and engineering. He has worked as Lecturer at M.M.M. Engg. College, Gorakhpur from 1991 to 1996. He joined Thapar Institute of Engineering & Technology in 1996 as a Systems Analyst in the Computer Centre and is presently associated with the same Institute. He has been a visiting faculty to many institutions. He has published over 100 papers in referred journals and conferences (India and Abroad). He is a MISCI(Turkey), LMCSI (Mumbai), GMAIMA (New Delhi). He is a certified software quality auditor by MoCIT, Govt. of India. His research interests include wireless networks, routing algorithms and securing ad hoc networks and data mining.

Dr. Ram Bahadur Patel is currently working as Associate Professor & Head at Department of Computer Science & Engineering, G. B. Pant Engineering College, Ghurdauri(Pauri-Garhwal). He received PhD from IIT Roorkee in Computer Science & Engineering, PDF from Highest Institute of Education, Science & Technology (HIEST), Athens, Greece, MS (Software Systems) from BITS Pilani and B. E. in Computer Engineering from M. M. M. Engineering College, Gorakhpur, UP. Dr. Patel is in teaching and Research & Development since 1991. He has supervised 36 M. Tech, 7 M. Phil. and 8 PhD Thesis. He is currently supervising 6 PhD students. He has published 130 research papers in International/National Journals and Refereed International Conferences. He has written 7 text books for engineering courses. These books are being followed prominently in all Under Graduate/Post Graduate courses and also for Research, in various Technical Institutions/Universities in India. He is member of ISTE (New Delhi), IEEE(USA). He is a member of various International Technical Committees and participating frequently in International Technical Committees in India and abroad. His current research interests are in Mobile & Distributed Computing, Mobile Agent Security and Fault Tolerance, Sensor Network etc.