School of Computing Science,
University of Newcastle upon Tyne

A Grid-based System for Microbial Genome Comparison and Analysis

Sun Y., Wipat A., Pocock M., Lee P., Watson P.,
Flanagan K. and Worthington J.

Technical Report Series

CS-TR-881

December 2004

A Grid-based System for Microbial Genome Comparison and Analysis

Yudong Sun, Anil Wipat, Matthew Pocock, Pete A. Lee, Paul Watson, Keith Flanagan and James T. Worthington
School of Computing Science, University of Newcastle upon Tyne, UK
{*Yudong.Sun, Anil.Wipat, Matthew.Pocock, P.A.Lee, Paul.Watson, Keith.Flanagan, j.t.worthington*}*@ncl.ac.uk*

Abstract

Genome comparison and analysis can reveal the structures and functions of genome sequences of different species. As more genomes are sequenced, genomic data sources are increasing in size and availability such that their analysis is beyond the processing capabilities of most research institutes. The Grid is a powerful solution to support large-scale genomic data processing and genome analysis. This paper presents the Microbase project that is developing a Grid-based system for genome comparison and analysis, and discusses the first implementation of the system (called MicrobaseLite). MicrobaseLite uses a scalable computing environment to support computationally intensive microbial genome comparison and analysis, employing state-of-the-art technologies of Web Services, notification, comparative genomics and parallel computing. Microbase will support not only system-defined genome comparison and analysis but also user-defined, remotely conceived genome analysis.

1. Introduction

Genome sequences provide abundant information about species from microorganisms to human beings. The comparison and analysis of genome sequences (including nucleotides and proteins) allows us to investigate genome structure and make predictions about the functions and activities of organisms [1]. Genome analysis can enhance our understanding of life science, and the discoveries in genome analysis can drive advances in medicine, agriculture and other sciences and technologies.

One application of genome comparison and analysis is in the design of therapeutic drugs. For example, say a new bacterium is found to cause a severe disease in humans. Scientists can experimentally determine the genome sequence of the bacterium. As proteins determine the functions and activities of an organism, the protein sequences of the new bacterial genome can be compared against the databases of all known bacterial genomes and even higher mammals' genomes to find the relationship between the new genome and the known genomes. This comparative analysis can identify proteins unique to the new bacterium that may be the target for the design of new antibacterial drugs [2].

In such an application, large genome databases will need to be searched and extensive comparison and analyses performed. To date, large genome databases have been established to accommodate genome data for public use such as EMBL (European Molecular Biology Laboratory) database [3], GenBank [4] , UniProt (Universal Protein Resource)/Swiss-Prot [5], PDB (Protein Data Bank) [6], and PIR (Protein Information Resource) [7]. Genome databases are experiencing rapid expansion as the rate of complete genome sequencing is continually increasing. This advancement presents a growing need for effective storage and querying approaches to the genome data.

With the accumulation of genome data, genome comparison and analysis has become a data-intensive and compute-intensive task. Many tools have been developed to perform genome comparison and analysis in different ways. The BLAST programs [8] are widely used tools for searching protein and nucleotide (DNA) databases to identify sequence similarities by performing local alignment between a query sequence and each of the sequences in a database. The BLAST family includes a number of variants. For example, BLASTP is a standard protein-protein comparison tool; BLASTN is for nucleotide-nucleotide comparison; BLASTX translates a nucleotide sequence to proteins that are compared against a protein database, and PSI-BLAST can find very distantly related proteins by a two-round protein-protein comparison. MUMmer [9] is a fast comparison tool that can rapidly align two large nucleotide sequences using a suffix-tree based algorithm. PROmer is a variant of MUMmer that generates the protein-level alignment for two nucleotide sequences based on the translation of nucleotide sequence to proteins. Ssearch [10] is a rigorous comparison program for global similarity between a database of sequences and a query sequence using the Smith-Waterman method [11] which is an extremely time-consuming algorithm.

These recent developments in biology and bioinformatics present a considerable challenge to the efficient management of genomic data sources and the high-performance systems for genome analysis. Grid computing has been proposed as a potential solution to these requirements [12-14]. The Grid can be used to integrate genome data sources and computing resources to build integrated genome databases and powerful computing platforms for genomic data processing, in particular for genome analysis and comparative genomics. As the Grid is a new technology for genome analysis, only a limited number of projects have been reported in this field.

The Microbase project aims to develop a Grid-based system to support large-scale genome comparison and analysis, in response to the influx of new genomes, by harnessing the data resources and computing resources on the Grid. As the first prototype developed by the project, Microbase*Lite* provides a pre-computed dataset of all-against-all microbial genome comparisons generated by a suite of genome comparison tools, and creates a scalable computing environment to perform computationally intensive genome comparison and analysis. Based on the pre-computed dataset, extensive genome analysis can be conducted, for example, to discover the homologues (including orthologues and paralogues) of the genes. The pre-computed dataset is dynamically integrated with an authoritative genome data source, the EMBL database [3]. When new genomes are published there, a Web Services based notification mechanism is used by Microbase*Lite* to automatically import the new genome data and compare the new genomes against all existing genomes, to update the pre-computed dataset. A task scheduler has been developed that assigns this large number of genome comparison jobs to run on Grid resources to accelerate the execution process. Microbase*Lite* acts as a prototype for gathering further requirements from biology and bioinformatics community to improve the design and implementation of the complete Microbase system.

The analysis of the large volume of genome data usually exceeds the computing resources in individual research institutes. The Microbase project will ultimately provide a Web Services interface for external clients to submit user-defined, remotely conceived genome analyses that can access and use the pre-computed dataset. It will accept and run the user-submitted algorithms on Grid resources on behalf of the clients.

The Microbase project also concerns the Grid-based approaches to interpret the genome comparison results in the context of a range of relevant biological principles including gene expression, protein function, metabolic pathways, and taxonomy. An object-based tool OGRE is being designed to represent and analyse genome rearrangement features in a formally defined ontology.

The rest of this paper is organised as follows. Section 2 introduces the related work. Section 3 describes the Microbase*Lite* architecture. Section 4 discusses the implementation and performance of Microbase*Lite*. Section 5 presents a use case, and Section 6 gives the conclusions and future work.

2. Related Work

Grid-based technologies are on the frontier of comparative genome analysis, although a number of projects have been undertaken in large research centres.

PUMA [15] is an integrated computational framework developed by the computational biology group of Argonne National Laboratory. It uses Grid technology to support high-throughput analysis of genomes for comparative and evolutionary analysis of metabolic processes on various levels of biological organisation in the context of phenotypic and taxonomic information derived from authoritative sources.

The TIGR Grid project [16] provides an in-house repository of protein and nucleotide data made available by major genome data repositories such as GenBank, PIR, and Swiss-Prot. In order to create non-redundant protein databases for annotation (i.e., identifying the features of a genome sequence), TIGR performs an all-against-all search on all proteins from these sources to create clusters of similar proteins. The data set of proteins is partitioned into multiple subsets and runs BLAST searches in parallel on an in-house Grid using Condor [17].

GADU (Genome Analysis and Database Update tool) [18] is a collaborative project between the Globus project and the Argonne bioinformatics group that has developed an automated, high-performance, scalable computational pipeline for data acquisition and analysis of newly sequenced genomes, based on DOE Science Grid backend [19].

GPSA (Grid Protein Sequence Analysis) [20] provides a web portal that allows users to submit protein sequences for homology searches. Users can select among BLAST, FASTA [10] or other tools to run the search. GPSA will dispatch the tool to run on the Grid infrastructure provided by the EGEE (Enabling Grids for E-science in Europe) project [21] to search the homologous sequences against certain databases.

Compared to this related work, Microbase*Lite* features a pre-computed dataset generated by a suite of genome comparison tools, that reveal the similarities of genome sequences at different levels. This dataset allows users to efficiently conduct further genome analyses without the need to regenerate the vast volume of data. Microbase*Lite* also provides a notification service for automatic dataset update. Furthermore, the Microbase project will be unique with its support for users to submit their own genome analysis computations. It will accept user-defined algorithms through a Web Services interface to be executed using its pre-computed dataset and Grid resources, enabling biologists to conduct extensive genome analysis using customised algorithms.

3. Microbase*Lite* Architecture

Microbase*Lite* is composed of distinct components deployed at the server side or client side. The components interoperate through Web Services interfaces and their operation is orchestrated through a Web Service-based notification mechanism.

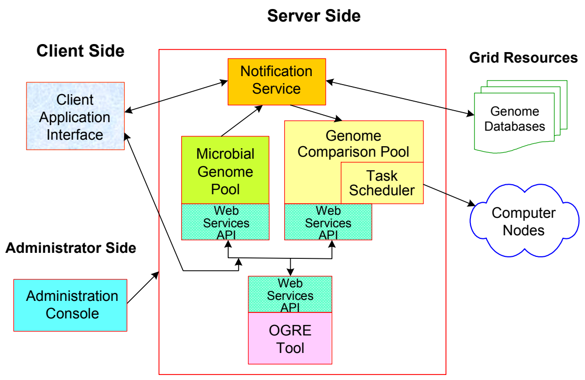

Figure 1. The Microbase*Lite* architecture

As Figure 1 shows, the server-side components include the Microbial Genome Pool, the Genome Comparison Pool, the Notification Service, and the OGRE tool (discussed below). Web Services interfaces are provided by each component to support interoperability between the components and provide services to clients. The client-side component is the Client Application Interface. There is also an administrator component for system administration purpose.

3.1. Microbial Genome Pool

The Microbial Genome Pool (or *the genome pool*) shown in Figure 2 provides a local database of complete microbial genome sequences. The genome pool collects the microbial genomes published at an authoritative genome data source (at present, the EMBL database). Genome data is stored in the local microbial genome database and used as the data source for the genome comparisons performed in the Genome Comparison Pool (see Section 3.2).

The local database is automatically updated when new genomes are published in the EMBL database. The update is activated by the Notification Service which will be discussed in Section 3.4. When a new genome is published, the Notification Service sends a notification message to the genome loader. The notification initiates the genome loader to download the new genome file from that site, and then parses and loads the genome sequence into the local database. The genome pool also sends a notification to subscribed clients to announce the new genome sequence.

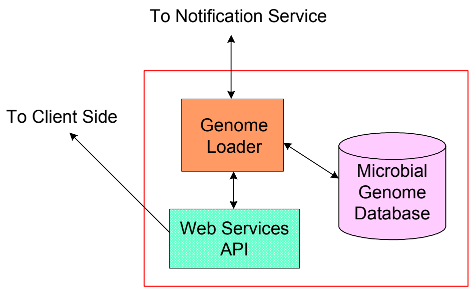

Figure 2. The Microbial Genome Pool

The Web Services API of the genome pool allows remote users to access the microbial genome database. Users can flexibly query the nucleotide and protein sequence information as well as the annotations and features of the genomes.

3.2. Genome Comparison Pool

The Genome Comparison Pool (or *the comparison pool*) shown in Figure 3 is a core component. It performs all-against-all comparisons for the genomes loaded in the Microbial Genome Pool, and populates the comparison results into the comparison database as the pre-computed dataset. The all-against-all genome comparison is performed using a suite of tools including BLASTP, BLASTN, MUMmer, PROmer and Ssearch to identify the similarities of the genomes at different levels. MSPcrunch [22] is also used as the post-processing for the BLASTN results to filter the most relevant data. The comparison database provides the pre-computed genome comparison dataset for users to browser the similarities of the genomes and to perform further genome analysis.

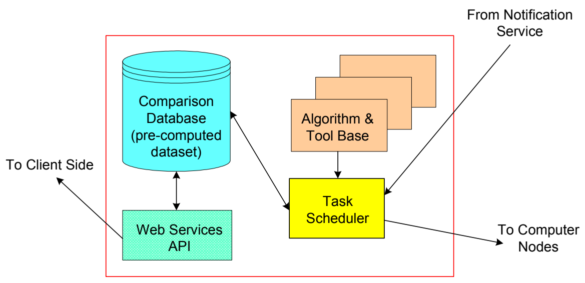

Figure 3. The Genome Comparison Pool

The all-against-all genome comparison generates a large number of computations. The Microbial Genome Pool currently has 165 genomes loaded in the local database, 137 of which are bacterial genomes. The length of a bacterial genome sequence is typically in the order of 10^6 bps (nucleotide base pairs) and 10^3 proteins; each protein is approximately 200-400 amino acids long. In total, 163,350 (=165×165×6) comparisons need to be performed for these genomes with the six tools. The majority of the comparisons are computationally intensive, particularly for BLASTP and Ssearch. For example, the BLASTP comparison between two bacteria: *Bacillus cereus* and *Bacillus anthracis*, requires two input files of 1.5MB each and produces 95MB output data. The comparison takes 12 minutes on a 2.8GHz Intel Xeon processor. Consequently, the overall execution of the all-against-all comparison is extremely time-consuming. Hence, Grid resources are needed to run the comparisons simultaneously. A task scheduler has been developed to support the parallel execution of the comparisons on the Grid or on a cluster of computers. Section 3.3 will discuss the task scheduler in detail.

The comparison pool provides a parser for each of the comparison tools to analyse the comparison output produced by the tool. When a comparison is completed, a corresponding parser is invoked to extract the required data from the raw output. The comparison database is then populated with the extracted data to form the pre-computed dataset for further use.

The Web Services API of the comparison pool allows external users to access the pre-computed dataset, for example, retrieving the protein similarities between two genomes generated by the BLASTP result. In next stage of the project, the Web Services API will be enhanced to allow users to submit user-defined genome analysis algorithms that operate on the pre-computed dataset. The task scheduler will manage the execution of the algorithms. The results will be returned to the users through the Web Services interface. User-defined algorithms will be archived in the algorithm and tool base for reuse. A use case will be discussed in Section 5 that identifies COGs (clusters of orthologous groups) of proteins using the pre-computed BLASTP results.

3.3. Task Scheduler

In the comparison pool, a task scheduler has been developed to support the parallel execution of the comparison jobs on a networked system such as a cluster of computers or the Grid. The task scheduler calls a general-purpose job management middleware, e.g. N1 Grid Engine (formerly Sun Grid Engine) [23], to submit the comparison jobs for execution. A comparison job is submitted by calling the job submission command of the middleware. The latter in turn allocates a computer node to run the job. A genome comparison job includes the following operations: retrieving the genome sequences from local database; running a tool to compare the sequences; invoking a parser to analyse the comparison results; and loading the result into the comparison database. Figure 4 shows the framework of the task scheduler.

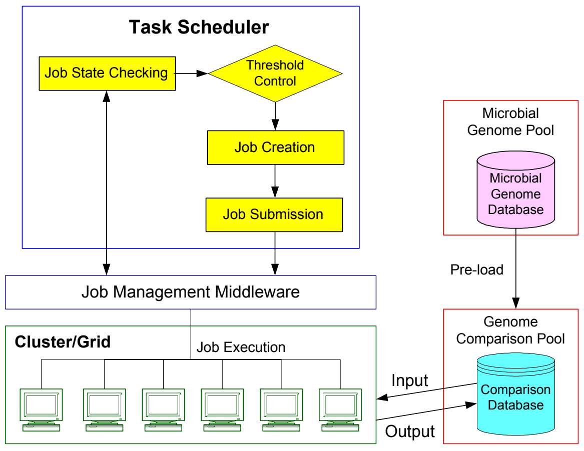

Figure 4. The task scheduler

The task scheduler coordinates the whole execution procedure of all jobs. As hundreds of thousands comparison jobs can be created, the task scheduler applies a threshold control to job submission in order to prevent submitted jobs from overwhelming the system. Under the threshold control, the total number of running jobs will not exceed the capacity of the available computer nodes. The task scheduler continually checks the states of the running jobs. Once a job has completed, the task scheduler immediately submits a new job to be executed. The task scheduler will terminate the whole comparison procedure when all jobs have completed.

The execution time of a genome comparison depends on the length of the sequences and the complexity of a comparison algorithm. The time varies significantly between different comparisons. As presented in Section 3.2, the BLASTP comparison of *Bacillus cereus* and *Bacillus anthracis* takes 12 minutes, whereas the MUMmer comparison for the same sequences takes only 22 seconds. Along with the underlying job management middleware, the task scheduler can maintain the asynchronous execution of the comparison jobs which have different execution times. Subsequent jobs are gradually submitted for execution in accordance with the completion of preceding jobs. Hence, the workload of the comparisons can be dynamically distributed to the computer nodes and the overall execution time of all-against-all genome comparison can be minimized.

3.4. Notification Service

The Notification Service in Microbase*Lite* is implemented using the ^{my}Grid notification system [24]. The ^{my}Grid notification is a Web Service based system for event notification that supports topic-based publisher and subscriber messaging, push and pull notification model, and asynchronous delivery. Clients can subscribe to receive notification messages on a registered topic. A client can be a user or a software component. The push model delivers a notification by calling back to client code deployed as a Web Service at the client side, or sending an email to a registered address. Microbase*Lite* uses the push model with client code call-back to notify users of the availability of a new genome - the Microbial Genome Pool uses this notification to update the local genome database. There is a probe deployed to periodically check the EMBL database. When a new microbial genome is published there, the probe will push a notification message to the genome loader of the genome pool. The notification triggers the genome loader to download the new genome file and store the genome sequence into the local database after parsing. The genome pool can also notify the comparison pool to activate the task scheduler to start the comparison of the new genome against all previously loaded genomes in the local database and hence to update the pre-computed dataset. The Notification Service also allows external clients to subscribe to the notification of new genome. A notification message will thus be sent to the clients when Microbase*Lite* has loaded a new genome.

3.5. OGRE

A further novel component provided by the Microbase system is a research tool called OGRE (Object based Genome REarrangements). Genome rearrangements such as insertions, deletions, and inversions can be visualised by existing tools [25, 26]. OGRE intends to develop a formally defined set of terms relating to genome rearrangement in the form of ontology. Formal definitions can be rigorously checked to ensure that they are logically consistent. The aim of OGRE is to use these definitions as a basis on which to develop an object-oriented data model and algorithms for the comparison and analysis of genome sequences. OGRE is a sister project to Microbase, but will be fully integrated; OGRE will provide a service interface to facilitate integration with Microbase*Lite* or other tools. The ontology used to describe genome rearrangements is currently under development, and there is a working prototype capable of detecting some simple features.

3.6. Client Application Interface

The Client Application Interface is built on the Web Services APIs of the server-side components for remote users to easily access the pre-computed dataset and associated information provided by Microbase*Lite*. The client interface provides users with an integrated view of the data stored in different components, such as the pre-computed dataset in the Genome Comparison Pool and the genome sequences in the Microbial Genome Pool, and supports the cross-reference of related data. The interface allows users to submit queries to Microbase*Lite* by specifying the parameters of query and reference genomes, which comparison tool to use and the range of comparison results, for example, the BLASTN result between two genomes. The interface calls the Web Services APIs to retrieve the required data from the databases in Microbase*Lit*e and returns the data to the users which can be displayed in graphic or textual format depending on the type of information to be shown. The interface also provides the cross-references of a query by which users can find further information associated to the query such as the detailed description of the compared genome sequences, the description of a comparison tool and the hyperlinks to related genome databases. As future work, the client interface will allow users to submit their own comparisons for execution and receive the results.

4. Microbase*Lite* Implementation

Microbase*Lite* has implemented the components discussed in Section 3. The Microbial Genome Pool takes the genome data in EMBL files from the EMBL database. It uses BioJava API [27] to parse the EMBL files and load the genome sequences into the microbial genome database that is a PostgreSQL database with the BioSQL schema [27]. The Web Services API is implemented using Apache Tomcat and Axis.

At present, the Genome Comparison Pool runs the comparison jobs on a cluster of dual-processor computers. The task scheduler is based on N1 Grid Engine which is middleware that can build and manage Grid resources and allows users to access the Grid. With the support of the N1 Grid Engine or other Grid middleware, we plan to extend the computing system to the Grid. The pre-computed comparison dataset is stored in a MySQL database. The all-against-all comparison has been performed amongst the 165 genomes, and a 16GB dataset of comparison results has been generated.

The Client Application Interface is implemented as graphical user interface (GUI) that calls the Web Services APIs of the server-side components. Users can query the data provided by Microbase*Lite* through this interface. Firstly, the interface presents users with a selection panel that displays a complete list of reference and query genomes and comparison tools available in Microbase*Lite*. Users can query any genome comparison result by selecting a pair of reference and query genomes and choosing a tool in the list. The query is then sent to Microbase*Lite,* and the required data is retrieved from the pre-computed dataset and returned to the client side via the Web Services API. The result of the query is displayed to the users in the graphical browser as shown in Figure 5, which depicts the BLASTN result between two genomes. The graph shows the similarities between the nucleotide sequences and uses arrows to highlight the genes that encode genome features including proteins, tRNA, mRNA, etc. Users can flexibly zoom in the graph to view the details of the sequences by sliding the resolution scale on the right side. Users can also click on the arrows to get the cross-references of detailed information of the features.

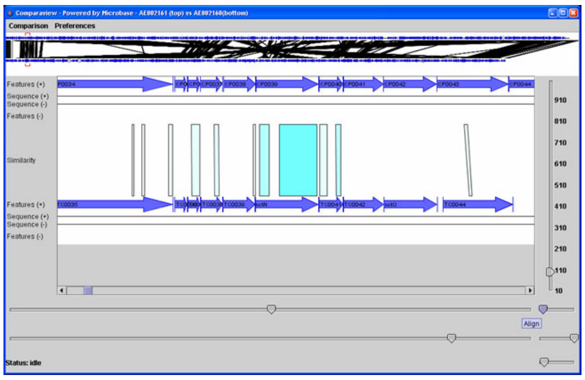

Figure 5. The client application interface for genome comparison results

Microbase*Lite* is available for external use. A description of the service APIs can be found at http://vindaloo.ncl.ac.uk:8090/microbase/index.html.

4.1. Performance

The performance test of Microbase*Lite* is focused on the scalability of the system in supporting large-scale genome comparison. The scalability is tested by running all-against-all comparisons on a cluster of computer nodes; each node contains dual 2.8GHz Xeon processors. The N1 Grid Engine is installed on the cluster. In the test, all-against-all comparison is executed amongst a group of genomes selected from the local database, using five tools (BLASTP, BLASTN, MSPcrunch, MUMmer and PROmer). As Ssearch is an extremely slow program, it will be separately executed later. The task scheduler manages the parallel execution of the comparison jobs.

Table 1. Execution times of all-against-all genome comparison (minutes)

Processors / *Genomes*	1	10	20	30	40
10 genomes	978.0	103.0	57.7	48.5	37.3
20 genomes	2387.7	251.5	147.0	116.1	94.5
30 genomes	5064.5	533.5	295.7	226.2	178.5

Scalability is assessed by the execution time and speedup obtained in running the all-against-all comparison on different numbers of genomes and different number of processors. Table 1 shows the execution times of these comparisons which include both the execution time of the tools and the time of data loading and analysis. Figure 6 shows the speedup achieved.

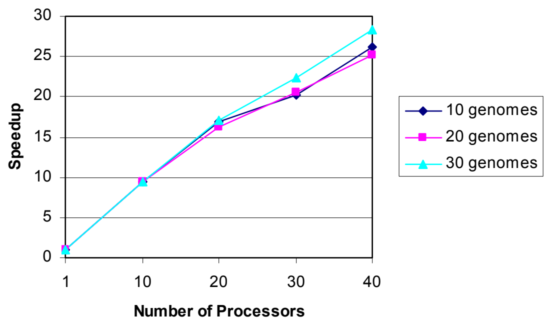

Figure 6. Speedup of all-against-all genome comparison

The test shows that useful speedup can be achieved when employing more processors to run the comparisons. The best speedup appears in the case of the largest dataset, 30 genomes. The performance verifies that Microbase*Lite* has demonstrated satisfactory scalability in running all-against-all genome comparison with the support of the task scheduler. The all-against-all comparison between 165 genomes using the five tools has also been tested on 40 processors. The entire execution time was 68 hours. This is an encouraging time scale for such a large-scale comparison. Considering the Grid computing support provided by the underlying N1 Grid Engine, better performance can be expected when the computing environment is extended to utilise Grid resources.

5. Use Case

The pre-computed dataset of Microbase*Lite* has been used to search for COGs (clusters of orthologous groups) in proteins. The relationships of the proteins from different genomes can be classified by the homologues (i.e. the similarities) including orthologues and paralogues. Paralogues are homologous proteins in a same genome. Orthologues are homologous proteins in different genomes that evolved from a common ancestral gene. Orthologues often retain the same function in the process of evolution. The identification of orthologues is an important methodology for the prediction of the functions of a protein or a group of proteins, in particular for newly sequenced genomes [28]. The orthologues and paralogues can be identified based on the similarities of the proteins found by genome comparison.

We use the COG construction algorithm proposed by the COG database project [28-30] to identify COGs using the BLASTP results provided by the pre-computed dataset. The best hits, that is the most similar proteins, are extracted from the BLASTP results based on the similarity score. The COGs can be identified from the best hits. The algorithm has identified 8945 COGs which consist of 24832 different proteins among the 165 genomes in Microbase*Lite*. At the same time, the algorithm has found 8045 paralogue groups that include 18583 different proteins. These results can be used to reveal the evolutionary relationships of the proteins in the genomes.

6. Conclusions

The Microbase project will exploit a Grid-based environment to support both biologists and bioinformaticians in carrying out comprehensive genome comparison and analysis. The Microbase*Lite* prototype system has provided a pre-computed dataset of microbial genome comparisons integrated with the service-based API to access it. The ultimate goal of Microbase is to provide a remotely accessible system to perform user-defined genome analysis. The future work will concentrate on two issues. First, the Microbase system will implement seamless integration with Grid resources to meet the computational and data requirements of analysing an almost exponential influx of new genomes. Second, Microbase will support user-defined, remotely conceived genome analysis. For this purpose, a workflow framework is a promising model to enable users to define, submit, and enact genome analysis algorithms. The Taverna project supports Grid-based workflows [31] that potentially provides useful mechanisms to implement the user workflow submission and enactment in the Microbase system.

Acknowledgments

The Microbase project is supported by the BBSRC e-Science and Bioinformatics initiative and the DTI (Grant number 13/BEP17027). We gratefully acknowledge the support of the North-East Regional e-Science Centre, UK.

References

[1] A. K. Bansal and T. E. Meyer, "Evolutionary analysis by whole-genome comparisons," *Journal of Bacteriology*, vol. 184, no. 8, 2002, pp. 2260-2272.

[2] A. M. Lesk, *Introduction to Bioinformatics*. Oxford University Press, Oxford, 2002.

[3] EMBL nucleotide sequence database, http://www.ebi.ac.uk/embl/index.html

[4] GenBank overview, http://www.ncbi.nlm.nih.gov/Genbank/GenbankOverview.html

[5] UniProt: the universal protein resource, http://www.ebi.uniprot.org/index.shtml

[6] PDB: the protein data bank, http://www.rcsb.org/pdb/

[7] PIR: the protein information resource, http://pir.georgetown.edu/

[8] S. Altschul, T. Madden, A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, 1997, pp. 3389-3402.

[9] S. Kurtz, A. Phillippy, A. Delcher, M. Smoot, M. Shumway, C. Antonescu, and S. Salzberg, "Versatile and open software for comparing large genomes," *Genome Biology*, vol. 5, no. 2, 2004, pp. R12.

[10] W. R. Pearson, "Empirical statistical estimates for sequence similarity searches," *Journal of Molecular Biology*, vol. 276, no. 1, 1998, pp. 71-84.

[11] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, 1981, pp. 195-197.

[12] R. Stevens, A. Robinson, and C. Goble, "myGrid: personalised bioinformatics on the information grid," *Bioinformatics*, vol. 19 Suppl 1, 2003, pp. i302-i304.

[13] E. Huedo, U. Bastolla, R. Montero, and I. Llorente, "Computational proteomics on the Grid," *New Generation Computing*, vol. 22, no. 2, 2004, pp. 191-192.

[14] A. Konagaya, F. Konishi, M. Hatakeyama, and K. Satou, "The superstructure toward open bioinformatics grid," *New Generation Computing*, vol. 22, no. 2, 2004, pp. 167-176.

[15] M. D'Souza, J. Huan, S. Sutton, M. Romine, and N. Maltsev, "PUMA2 - An environment for comparative analysis of metabolic subsystems and automated reconstruction of metabolism of microbial consortia and individual organisms from sequence data," Report, ANL/MCS-TM-240, 1999.

[16] TIGR Grid Computing, http://www.tigr.org/grid/

[17] Condor high throughput computing, http://www.cs.wisc.edu/condor/

[18] A. Rodriguez, D. Sulakhe, E. Marland, V. Nefedova, G. Yu, and N. Maltsev, "GADU - Genome analysis and database update pipeline," Report, ANL/MCS-P1029-0203, 2003.

[19] DOE ScienceGrid: summary of progress, Feb. 2002 to Feb. 2003, http://www.doesciencegrid.org/Grid/management

[20] GPSA: Grid protein sequence analysis, http://gpsa.ibcp.fr/

[21] EGEE: enabling grids for e-science in Europe, http://public.eu-egee.org/

[22] E. L. Sonnhammer and R. Durbin, "MSPcrunch: a blast enhancement tool for large-scale sequence similarity analysis," Report, 1997.

[23] N1 Grid Engine 6 user's guide, http://docs.sun.com/db/doc/817-6117

[24] A. Krishna, V. Tan, R. Lawley, S. Miles, and L. Moreau, "myGrid notification service," presented at UK e-Science All Hands Meeting, Nottingham, 2-4 September 2003.

[25] ACT: Artemis comparison tool, http://www.sanger.ac.uk/Software/ACT/

[26] J. Yang, J. Wang, Z. Yao, Q. Jin, Y. Shen, and R. Chen, "GenomeComp: a visualization tool for microbial genome comparison," *Journal of Microbiological Methods*, vol. 54, no. 3, 2003, pp. 423-426.

[27] BioJava, http://www.biojava.org/

[28] R. L. Tatusov, E. V. Koonin, and D. J. Lipman, "A genomic perspective on protein families," *Science*, vol. 278, 1997, pp. 631-637.

[29] R. L. Tatusov, M. Y. Galperin, D. A. Natale, and E. V. Koonin, "The COG database: a tool for genome-scale analysis of protein functions and evolution," *Nucleic Acids Research*, vol. 28, no. 1, 2000, pp. 33-36.

[30] COGs, phylogenetic classification of proteins encoded in complete genomes, http://www.ncbi.nlm.nih.gov/COG/

[31] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Greenwood, T. Carver, M. Pocock, A. Wipat, and P. Li, "Taverna: a tool for the composition and enactment of bioinformatics workflows," *Bioinformatics*, 2004.