School of Computing Science,
University of Newcastle upon Tyne

Exploring Microbial Genome Sequences to Identify Protein Families on the Grid

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

Technical Report Series

CS-TR-931

October 2005

Exploring Microbial Genome Sequences to Identify Protein Families on the Grid

Yudong Sun, Anil Wipat, Matthew Pocock, Peter A. Lee, Keith Flanagan, and James T. Worthington

***Abstract*—The analysis of microbial genome sequences can identify protein families that provide potential drug targets for new antibiotics. With the rapid accumulation of newly sequenced genomes, the analysis of complete genome sequences has become a computationally- and data-intensive problem which is intractable on common computer systems. This paper presents the Microbase project that has developed a Grid-based system to support large-scale comparative analysis of complete microbial genome sequences, and the identification of protein families based on the analysis. The system integrates Grid computing with genomic databases to provide a high-performance environment for efficient genome comparison, analysis and protein family search. A pre-computed dataset of sequence similarities and homologous protein families has been generated which can assist the discovery of new therapeutic agents and provide leads for drug development.**

***Index Terms*—Genome analysis, Grid, microbial genomes, protein families.**

I. Introduction

DEVELOPMENTS in comparative genomics are helping to provide novel techniques for therapeutic anti-microbial drug discovery. The comparative analysis of complete microbial genome sequences can identify unique proteins and homologous protein families conserved in and between genomes, which can be screened in the search for new antibiotic targets [1]-[3]. Genome analysis has become a promising route for developing new antibiotics to tackle the increasing risks of infections in humans, such as the emergence of new bacterial pathogens, the spread of epidemic diseases, and the intensified resistance to existing antibiotics [2].

With the rapid increase in the availability of complete microbial genome sequences, the comparative analysis of whole microbial genomes has become a computationally- and data-intensive problem. For example, whole sequence alignment and homology searches need to perform numerous computational operations over a huge volume of genomic data. This computational load taxes the capability of most common computing systems. Grid computing has been recognized as a fast growing technology that can support the computational requirements of grand-challenge applications in biology, biomedicine and bioinformatics. The Grid integrates computer resources available on the Internet, in effect to form a giant computing system capable of supporting large applications such as complete genome sequence comparison and analysis [4]-[6].

The Microbase project has developed a Grid-based system to support the timely dynamic or 'on-demand' comparative analysis of microbial genome sequences. The system is able to generate a large pre-computed dataset of genome comparison results. The pre-computed dataset acts as a data repository of pairwise sequence similarities on which various genome analyses can subsequently be implemented. Similarity searches have been conducted on this resulting dataset to find protein families among bacterial genomes. A protein family conserved in a phylogenic group of bacteria can be considered as a potential target of broad-spectrum antibiotics, whereas a protein unique to a specific pathogenic bacterium can be used as the target of a narrow-spectrum drug. The Grid-based system and pre-computed dataset are available to the users in biological and biomedical communities who can efficiently fulfill large-scale genome analyses on the dataset without having to repeat the time-consuming genome comparisons.

The first implemented system of the project, Microbase*Lite* has been developed on a campus Grid to investigate the capability of a Grid-based computing environment in supporting large-scale genome comparison and analysis. This system has produced a large dataset of all-against-all comparisons for 250 microbial genomes, mainly bacteria. The pre-computed dataset can be regularly auto-updated by a Web Service-based notification service to incorporate new genome sequences. A similarity search among the 250 genomes has also been implemented on the system using different searching algorithms to identify putative orthologues and COGs (clusters of orthologues groups). The system has been developed with Web-based user accessibility as a prime concern. A graphical client interface has been developed to allow remote users to query and view the pre-computed genome comparison results and protein families via Web Service interfaces.

In the rest of the paper, Section II introduces the related work. Section III presents the Microbase*Lite* system and the pre-computed dataset of genome comparison produced.

Manuscript received October 18, 2005. This work is supported by the UK BBSRC e-Science and Bioinformatics initiative and the DTI under Grant 13/BEP17027.

The authors are with the School of Computing Science, University of Newcastle upon Tyne, Newcastle upon Tyne, UK, NE1 7RU (e-mail: yudong.sun@ncl.ac.uk; anil.wipat@ncl.ac.uk; matthew.pocock@ncl.ac.uk; p.a.lee@ncl.ac.uk; keith.flanagan@ncl.ac.uk; j.t.worthington@ncl.ac.uk).

Section IV discusses the derivation of protein families based on the pre-compute dataset. Section V concludes with future work.

II. Related Work

Grid computing is increasingly adopted for biological and biomedical research. A number of Grid-based systems are being developed to support comparative analysis of genome sequences. GNARE (Genome Analysis Research Environment) [6] is a Grid-based system to run genome analysis tools, mainly BLAST, with automated workflow generation and to provide an integrated database of genome sequences and analysis results for further analysis. TIGR's DCE (Distributed Computing Environment) [7] is an institutional Grid system to perform genome analysis, including BLAST, MUMmer, and HMMsearch, and to maintain an in-house repository of protein and nucleotide data and a protein database of all-vs.-all search to identify protein similarity. NC BioGrid [8] is a regional Grid infrastructure that integrates computing, data storage, and networking resources to gather genomic data from different sources and provides the data to research and education consortium in North Carolina, and to perform genomic data analysis including BLAST. The GPSA (Grid Protein Sequence Analysis) [9] web portal provides a user interface to run protein sequence analyses, including BLAST, FASTA, SSEARCH, and ClusterW, on the European EGEE Grid [10].

Orthologues searches are important applications in comparative genomics that identify similar proteins and genes from different genomes for functional and evolutionary studies. The COGs database [11]-[13] contains the clusters of orthologous proteins identified from different phylogenetic lineages and has become widely accepted for the annotation of proteins. *coli*BASE [14][15] is a database of *Escherichia coli, Shigella,* and *Salmonella*, reflecting the full diversity of *E. coli* and its relatives, which includes the putative orthologues found in these genomes. The e-Fungi project [5] has performed homologues analysis for fungal genomes using BLASTP and a Markov Chain Clustering (MCL) method to cluster protein families for phylogenetic and pathogenic analysis of fungi.

Drug discovery is an emerging application area of Grid computing. myGrid [16] is a service-based Grid middleware framework to manage the complex process of life science research. It has been used to enact workflows for the genetic analysis of microarray data on the Grid to discover the genes involved in a genetic disease (Graves disease) as a drug target. myGrid supports data management, new discovery notification, and provenance management in the drug discovery process [17]. The EGEE Grid has recently established a drug discovery project for the virtual screening of a large amount of data to find potential drugs to treat infectious diseases such as malaria [18].

Compared with the related work, the Microbase project has a clear target to support the analytical research of microbial genomes with recognition of their importance to medical science as well as to environment research and the bioscience industry. The system will be open to biological and biomedical communities to perform user-defined analyses. Using the system, a pre-computed dataset of microbial genome comparison results has been generated from which further genomic analysis can be realized. We have carried out protein family searches for a dataset of proteins from microbial organisms, by means of Grid computing. Microbase concentrates on, but is not limited to, the analysis of microbial genomes. The environment is also suitable to support the analysis of other genomes and other genomic applications.

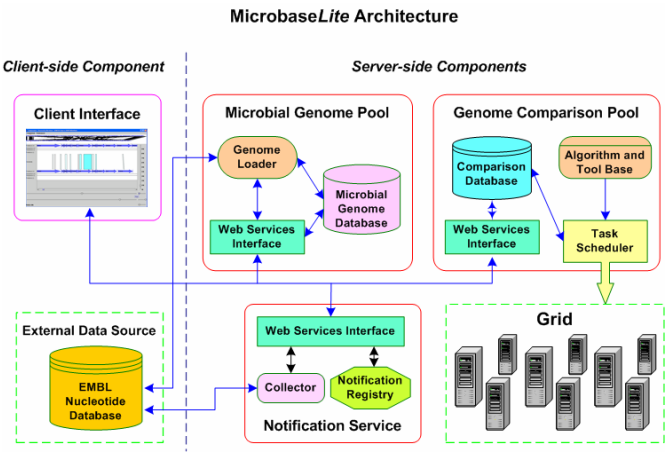

Fig. 1. Microbase*Lite* architecture includes (1) server-side components: microbial genome pool, genome comparison pool, and notification service; (2) client-side component: client interface. The Grid and EMBL nucleotide database are external resources.

III. MicrobaseLite

Microbase*Lite* is the first implementation in the Microbase project. Microbase*Lite* consists of multiple components that are integrated to service various comparative analyses of genome sequences. The major components are the microbial genome pool, the genome comparison pool, the notification service, and the client interface. The interoperation of the components is achieved through Web Service based interfaces and the components are orchestrated via Web Service based notifications. The interaction between the system and clients is also achieved via Web Services. Fig. 1 shows the architecture of Microbase*Lite*.

A. Microbial Genome Pool

The microbial genome pool maintains an up-to-date database of complete microbial genome sequences, most of which are bacterial genomes. Genomes published in the EMBL nucleotide database [19] are imported and loaded into the pool for use in later genome comparison.

To facilitate user access to the genome sequences, the microbial genome pool parses plain text EMBL records from the EMBL database using BioJava [20] and stores the sequence in the microbial genome database using the BioSQL

relational schema [21]—a schema for structural storage and retrieval of genome sequences. At the time of writing, the microbial genome pool holds 250 microbial genome sequences.

The microbial genome pool provides a Web Service based client interface for users to flexibly retrieve genome data from the microbial genome database. Using Java methods, a user can retrieve a DNA sequence, protein sequences, features (e.g. CDS, tRNA, mRNA), and annotations (e.g., a genome's ID, organism species, and references). A user can also retrieve a fragment of nucleotide sequence and query the features associated with that fragment.

The microbial genome pool can automatically update the local microbial genome database by a *notification service*. The notification service is a Web Service based mechanism for event notification, using the myGrid notification system [22][23]. In the notification service, a collector component is deployed to regularly check for new microbial genomes published in the EMBL nucleotide database. When a new genome is available, the collector sends a notification to trigger the genome loader of the microbial genome pool to load the new genome.

B. Genome Comparison Pool

The genome comparison pool is the central component responsible for conducting genome comparison and analysis on the Grid system, and for maintaining the results as a pre-computed dataset for user access. All-against-all comparisons have been performed for the 250 genome sequences loaded in the microbial genome pool.

The genome comparison pool uses four tools for pairwise comparison of the genome sequences: BLASTP, BLASTN, MUMmer, and PROmer. These tools are used to find the sequence similarities at nucleotide, protein or gene levels. BLASTP [24][25] is a protein-protein comparison tool that searches similar proteins between query and reference genomes. BLASTN [24][25] is a tool for pairwise alignment of nucleotide sequences to find similar nucleotide fragments. MUMmer [26][27] is a fast alignment tool for nucleotide sequences. It is employed, in addition to BLASTN, to get an abstraction of similar nucleotide fragments. PROmer [27] is a variant of MUMmer, which translates two nucleotide sequences into amino acid sequences in all six frames, finds all matches in the amino acid sequences, and then maps the matches back to the positions in original nucleotide sequences.

The 250 microbial genomes loaded in the microbial genome pool require 62,500 pairwise comparisons. A complete genome comparison is usually a computationally intensive task. For example, there are various pathogenic bacteria in genus *Bacillus*: *Bacillus anthracis* causes anthrax in humans and in animals, while *Bacillus cereus* causes food poisoning in humans. Using BLASTP to compare the protein sequences of the two species takes 12 minutes on a 2.8GHz CPU and produces 95MB output data. Another example is the comparison of the infectious bacteria in genus *Leptospira* that are causative agents of Weil's disease or canicola fever. Using BLASTN to compare the whole nucleotide sequences of *Leptospira interrogans serovar Lai str.* 56601 and *Leptospira interrogans serovar Copenhageni str.* Fiocruz L1-130 takes over 8 hours and produces 193MB output data. With the four comparison tools in use, the all-against-all comparison is inevitably an intensive job that exceeds the capability of common computing systems.

To handle this problem, the genome comparison pool exploits a Grid-based computing environment on which all-against-all genome comparison can be efficiently executed. In the execution, a pairwise comparison is specified as a comparison job. A large number of comparison jobs can be executed in parallel on the Grid system. A *task scheduler* has been designed to manage the parallel execution of jobs on the Grid system. The task scheduler creates the comparison jobs and submits the jobs to run on the Grid by means of a job submission middleware such as Globus Toolkit [28], Condor [29], or Sun ONE Grid Engine [30], depending on what middleware is available on the Grid system. The task scheduler controls the pace of jobs submission based on the usable hosts on the Grid system. A new job is submitted only when a running job has finished and a host has been vacated, to avoid congestion caused by a huge number of jobs concurrently occupying the system.

All comparison results are parsed and stored in a relational database, called the comparison database, for user access. As many applications of genome analysis are based on the similarities of genome sequences, the comparison database provides an instantly accessible data source to directly implement various in-depth genome analyses without the need to undertake the time-consuming genome comparisons. Section IV will discuss the search of protein families based on the BLASTP results provided by the pre-computed dataset.

At present, the Grid-based system is a campus Grid at our university. The 250-against-250 microbial genome comparisons have been completed and the comparison database occupies 28GB. Fig. 2 shows the performance of all-against-all genome comparisons of a selected number of microbial genomes. In Fig. 2(a) the execution time is the elapsed time of all pairwise comparisons using four tools, which includes the time for parsing and loading result data into the comparison database. The speedup in Fig. 2(b) is derived from the execution time. Fig. 2 demonstrates that good speedups can be achieved by exploiting significant numbers of CPUs.

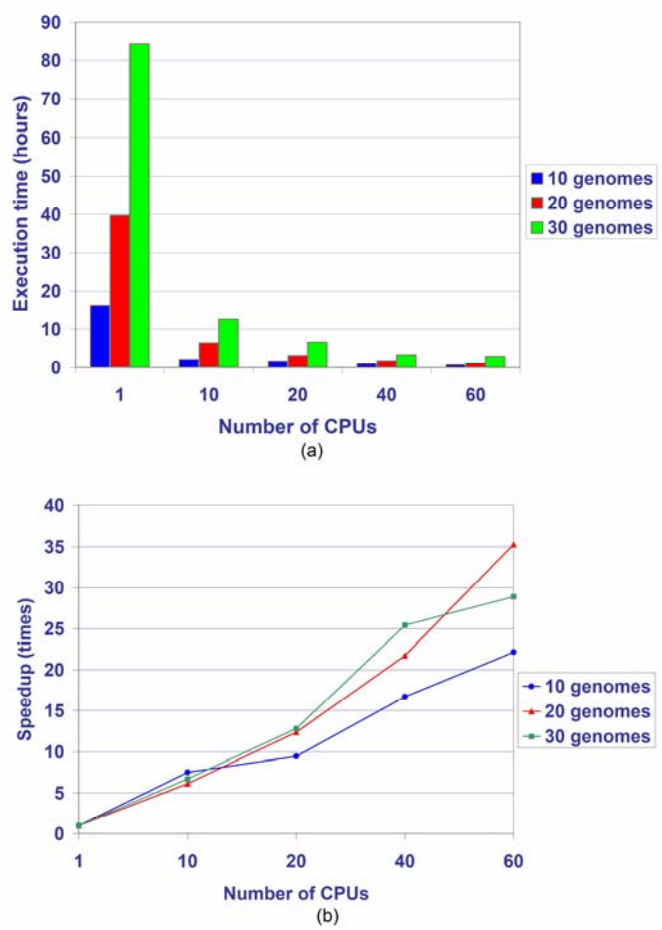

Fig. 2. Performance of all-against-all genome comparison on the Newcastle Campus Grid with varied number of genomes: (a) Execution time; (b) Speedup

New genome comparison results can be incrementally added to the comparison database. The notification service for updating the microbial genome database also triggers the genome comparison pool to update the comparison database. When a new genome is available, the task scheduler starts the comparison of the new genome against previously loaded genomes on the Grid system and updates the pre-computed dataset with new comparison results.

C. *Client Interface*

A client interface has been created to permit external users to access the genome sequence data and the pre-computed dataset in Microbase*Lite* via the Internet. The client interface provides an API (application programming interface) and a GUI (graphical user interface). The API is based on the Web Service interfaces using Apache Tomcat and Axis. User programs can call the API to retrieve data. The GUI has been developed on top of the API for interactive viewing of the genomic data and related information, and is deployed as a client on the user's system. The graphical interface presents a browser by which a user can submit queries to the server side. The query is sent to the server side and the requested data are retrieved from a database and returned to the client side via the Web Service interfaces, for display in the browser. Fig. 3

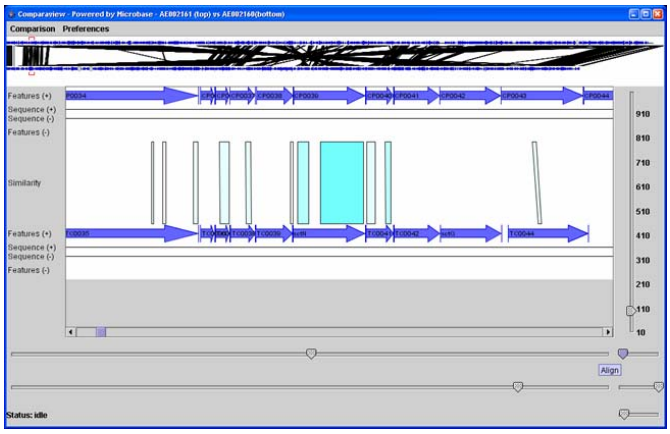

Fig. 3. The graphical client interface shows BLASTN alignment between *Leptospira interrogans serovar Lai str.* 56601 and *Leptospira interrogans serovar Copenhageni str.* Fiocruz L1-130. The whole sequence alignment is shown in the upper window. A segment of detailed alignment is displayed in the large window. Two horizontal arrowed lines represent two nucleotide sequences; each arrow represents a coding gene. Clicking on an arrow can pop up a window showing the encoded feature. The vertical bars in between indicate the similar fragments between two sequences. Sliding the rulers at the bottom can browse over the whole aligned sequences. Scrolling the scale on the right can zoom in or out on the alignment segment.

shows the graphical client interface displaying a segment of BLASTN alignment between the nucleotide sequences of *Leptospira interrogans serovar Lai str.* 56601 and *Leptospira interrogans serovar Copenhageni str.* Fiocruz L1-130. The current client interface provides a number of interactive facilities. For example, a user can browse other parts of the alignment or zoom into the alignment using the rulers on the panel. The arrows represent the coding genes on a nucleotide sequence. Clicking on an arrow will pop up a window showing the encoded feature. The user can also get the description of a genome, a comparison tool and related links through the browser.

IV. Protein Family Search

A protein family is a group of similar proteins. Proteins directly related to each other through evolutionary processes are called homologues, and can be further classified as 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. Similarity searches are an effective method to predict the evolutionary relations and infer the functions of a group of genes and proteins [12][13][31]. A group of orthologues can be considered as a potential target of broad-spectrum antibiotics.

Microbase*Lite* holds the pre-computed BLASTP results showing the pairwise similarities of proteins for the microbial genomes in the database. The similarity of proteins reflects the similarity of genes that encode the proteins. The dataset provides a foundation on which various homology searches can be implemented. The search of protein families has been

TABLE I
PUTATIVE ORTHOLOGUES OF ECS0014, *E. COLI* O157:H7 RIMD 0509952

Gene	Organism	Disease
dnaK (grpF, groP)	*Erwinia carotovora subsp. atroseptica* SCRI1043	A plant pathogen causing soft rot and blackleg in potato
dnaK	*Escherichia coli* CFT073	Urinary tract infections
dnaK	*Escherichia coli O157:H7* EDL933	Hemorrhagic colitis
dnaK	*Haemophilus ducreyi* 35000HP	Chancroid
dnaK	*Pasteurella multocida*	Pasteurellosis
dnaK	*Photorhabdus luminescens subsp. laumondii* TTO1	Toxemia and septicemia
dnaK	*Salmonella enterica subsp. enterica serovar Choleraesuis str.* SC-B67	Salmonellosis and swine paratyphoid
dnaK	*Salmonella enterica subsp. enterica serovar Paratypi A str* ATCC 9150	Paratyphoid fever
dnaK	*Salmonella enterica subsp. enterica serovar Typhi str.* CT18	Typhoid fever
dnaK	*Salmonella enterica subsp. enterica serovar Typhi Ty2*	Typhoid fever
dnaK	*Salmonella typhimurium* LT2	Gastroenteritis and food poisoning
dnaK	*Shigella flexneri 2a str.* 2457T	Dysentery
dnaK	*Shigella flexneri 2a str.* 301	Dysentery
dnaK	*Vibrio cholerae*	Cholera
dnaK	*Vibrio parahaemolyticus*	Gastroenteritis
dnaK	*Vibrio vulnificus* CMCP6	Gastroenteritis, wound infections and septicemia
dnaK	*Vibrio vulnificus* YJ016	Gastroenteritis, wound infections and septicemia
dnaK	*Yersinia pestis* CO92	Plague
dnaK	*Yersinia pestis* KIM	Plague
dnaK	*Yersinia pestis biovar Medievalis str.* 91001	Plague
dnaK	*Yersinia pseudotuberculosis* IP 32953	Gastroenteritis
dnaK	*Blochmannia floridanus*	Non-pathogen
dnaK	*Buchnera aphidicola str.* APS *(Acyrthosiphon pisum)*	Non-pathogen
dnaK	*Buchnera aphidicola str.* Sg *(Schizaphis graminum)*	Non-pathogen
dnaK	*Escherichia coli* K12	Non-pathogen
dnaK	*Mannheimia succiniciproducens* MBEL55E	Non-pathogen
PBPRA0697[a] (putative dnaK protein)	*Photobacterium profundum*	Non-pathogen
VF1467[a] (chaperone protein dnaK)	*Vibrio fischeri* ES114	Non-pathogen
VF1994[a] (chaperone protein dnaK)	*Vibrio fischeri* ES114	Non-pathogen

[a]Gene name is unavailable and locus tag is used instead.

implemented based on the BLASTP results, including the putative orthologues search and the COGs search.

A. Putative Orthologues

Putative orthologues are defined as the proteins that have mutual best hits in the BLASTP comparison with additional requirements on the aligned portions. We adopt the criteria specified by *coli*BASE [14][15] for the selection of putative orthologues. Protein families reflect the evolutionary relations of the genes that encode the proteins, so we use the terms "protein" and "gene" interchangeably when referring to orthologues in the following text. The search of putative orthologues starts on the filtering of mutual best hits in the all-against-all BLASTP results.

Definition 1: Given protein α from genome A and protein β from genome B (A and B are different genomes), α is a *best hit* to β if the hit has highest bit score and lowest E-value in all BLASTP hits between α and any proteins of genome B. The hit between α and β is a *mutual best hit* if α is a best hit to β and β is also a best hit to α.

The mutual best hit means that α and β are the most similar proteins among all proteins between genome A and B. The evolutionary and functional relationships between the similar proteins, and therefore the genes that encode the proteins, can be inferred based on the mutual best hits that are defined as putative orthologues.

Definition 2: If the mutual best hit between protein α and β satisfies two conditions on the aligned portion as following, α and β are *putative orthologues*:

1. The aligned portion has at least 80% amino acid identity.
2. The aligned portion covers at least 90% of the shorter sequence.

With the definitions, the search of putative orthologues is accomplished in three steps:

1. Filter out the best hits in all BLASTP hits of each protein against the proteins of each genome;
2. Filter out mutual best hits among the best hits;
3. Check the amino acid identity and alignment coverage of the mutual best hits to identify the putative orthologues that satisfy the two conditions in Definition 2.

Microbase*Lite* has a collection of 646,954 proteins from the 250 genomes. The pairwise BLASTP comparisons have reported more than 400 million hits. A parallel search has been performed to expedite the search process for the large dataset of proteins. Running on eight 2.8GHz CPUs, the search for putative orthologues was completed in ten days (it needs more than two months to run on a single CPU). No more CPUs have been used for the search because the search of best hits is a data-intensive process concentrating on examining a table of 400 million BLASTP hits with a total size of 22GB. The speed of parallel search is restricted by the speed of the database server—using more CPUs will not improve the speed of search. This problem can be solved by employing a distributed database scheme that supports parallel search. During the search, 287,490 proteins found putative orthologues representing 44.4% of the total proteins in our database. The putative orthologues reported by the search are dependent on the specified cutoff conditions of aligned portion. Using different cutoff percentages can increase or decrease the set of putative orthologues found in the search. Orthologues provide important information for varied biological researches such as evolutionary study and functional annotation, in addition to drug discovery [12]. Hence, the putative orthologues have been incorporated into the comparison database of genome comparison pool for biologists to use in their researches.

Our search is conducted on the 250 complete microbial genomes including different bacterial species. For example, our search has found 29 putative orthologues of the gene ECs0014 (dnaK), *Escherichia coli* O157:H7 RIMD 0509952, a pathogenic strain of *Escherichia coli* that causes severe food-poisoning disease. Table I shows the 29 putative orthologues of ECs0014 with associated bacterial organisms and diseases. As Table I shows, 72.4% of the 29 putative orthologues come from pathogenic bacterial species such as *Escherichia coli*, *Salmonella, Vibrios* and *Yersiniae* that can cause severe diseases in humans as well as in animals and plants. The putative orthologues provide useful information to find potential targets of new broad-spectrum antibiotics. In contrast, some genes are conserved in very limited number of organisms. In our dataset, 96,268 proteins have found only one putative orthologue respectively. For example, the gene fda, *Pseudomonas aeruginosa* PAO1 (a significant agent of bacteremia in burn victims, urinary-tract infections and hospital-acquired pneumonia) has only one putative orthologue, the gene fbaB of *Francisella tularensis subsp. tularensis* SCHU S4 that causes tularemia in humans and animals.

The mutual best hits found in the putative orthologues search can also be used for other homology search such as the COGs search.

B. COGs

COGs (Clusters of Orthologous Groups) are also a classification of homologous protein families [12][13]. Each COG is composed of orthologous proteins or orthologous groups of paralogous proteins from three or more genomes. The process of COGs search identifies both orthologous proteins from different genomes and paralogous proteins from the same genomes. The paralogues from a genome are gathered into a group that is considered as a single candidate orthologue in the search of COGs. Unlike the putative orthologues that only reflects one-to-many relationship of the proteins, COGs can reveal more comprehensive, many-to-many relationships amongst the proteins from the same and different genomes.

The search of COGs is based on the same set of mutual best hits obtained in the putative orthologues search. However, the COGs search does not set any cutoff requirement on aligned portions. In addition, the COGs search needs to identify all paralogues that are the mutual best hits from same genomes. Our COGs search is conducted by the following steps based the COGs construction protocol from the COGs database project [11]-[13]:

1. Filter out best hits and mutual best hits from BLASTP output (already done in the putative orthologues search).
2. Find paralogues in each genome and construct the groups of paralogues; replace individual paralogues with a unique ID per group of paralogues in the mutual best hits.
3. Search all groups of three orthologues among the mutual best hits. Given three proteins α, β, and γ, the proteins form a group of three orthologues if (α, β), (β, γ) and (α, γ) are mutual best hits. A group of paralogues is treated as a single orthologue in the formation of the groups.
4. Merge the groups that have at least a common mutual best hit if the merge will not gather the proteins from same genome (except those are paralogues) into a group.
5. The COGs are formed if the groups cannot be further merged.

The COGs search includes an exhaustive search of the three-orthologue groups and thereafter a continuous merge of the groups—a more compute-intensive and data-intensive method than the putative orthologues search. For a fast implementation of the COGs search, a *divide and conquer* [32] method is used to parallelize the search process. As Fig. 4 shows, the divide and conquer method of parallel COGs search consists of three phases:

1. *Divide*: divide the whole protein set into p subsets.
2. *Search*: search the groups of three orthologues for the proteins from each subset and perform an initial merge of the groups. This phase can be run in parallel on p processors.
3. *Merge*: merge the groups of orthologues generated from different subsets in $\log p$ rounds. Each round runs on a reduced number of processors to merge the groups of orthologues produced on different processors in

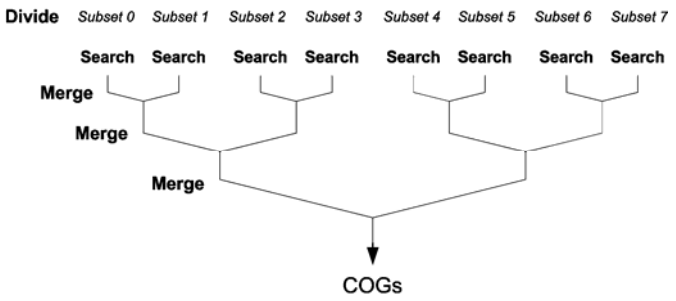

Fig. 4. Divide and conquer method for COGs search where p=8. The whole set of proteins are split into p subsets. The search of three-member orthologous groups is performed for each subset per processor, followed by $log\ p$ steps of merge.

pairs. The COGs are finally formed in the last round of merge which runs on one processor.

In the search of three-orthologue groups (α, β, γ), only the starting point α is selected from an associated subset. Its orthologues β and γ are searched in the whole protein set. Therefore, the divided search can find all groups of orthologues as a sequential search does.

The COGs search for all proteins of the 250 microbial genomes required 30 days on eight CPUs excluding the time for filtering mutual best hits which is already available for use. The search is estimated to require more than 200 days on a single processor. The search has identified 546,699 orthologues which consist of 531,441 single proteins and 15,258 groups of paralogues. In total, 571,701 proteins are assigned to one or more COGs that occupy 88.37% of the proteins from the 250 genomes. Also, 18,455 groups of paralogues have been found which consist of 47,608 proteins. The COG that includes the gene ECs0014 (dnaK), *Escherichia coli* O157:H7 RIMD 0509952 also includes the 29 putative orthologues of ECs0014 shown in Table I, in which 26 are individual proteins and 3 exist in paralogous groups. This result shows the consistency between the orthologues found by the COGs search and the putative orthologues search. Since a COG is formed by merging the orthologous groups, however, it assembles more orthologues together that reflect many-to-many relationships among proteins and genes. The COGs dataset is also incorporated into the genome comparison pool database to allow querying by users.

V. Conclusions

The Grid enables a more timely analysis of complete genome sequences and hence facilitates a more rapid and intensive exploration of the biological data they encode. In turn, this will allow knowledge to be more quickly derived in the face of rapidly accumulating genomic data. The Microbase project is developing technology to exploit Grid-based environments to support computationally intensive genome comparison and analysis, with a focus on the analysis of microbial genomes. Microbase*Lite* presented in this paper is a Grid-based system developed to support all-against-all comparison of complete microbial genome sequences. The pre-computed comparison results are useful for biological and biomedical researches to discover in-depth knowledge from the genomic data such as the identification of protein families. Protein families can be used to infer candidate targets for drug discovery as well as reveal the evolutionary relationship and functions of the proteins.

Future developments in Microbase will support user-defined, remotely conceived genome analyses. The system will enhance the ability to support user application submission and execution on the Grid system. A workflow framework will be used for the definition and enactment of user applications. More applications of genome analysis will be developed based on the pre-computed dataset such as metabolic reconstruction and promoter searches.

Acknowledgment

We gratefully acknowledge the support of the BBSRC, DTI (grant number 13/BEP17027) and the North-East Regional e-Science Centre, UK.

References

[1] M. T. Black and J. Hodgson, "Novel target sites in bacteria for overcoming antibiotic resistance," *Advanced Drug Delivery Reviews*, vol. 57, no. 10, 2005, pp. 1528-1538.
[2] H. Loferer, "Mining bacterial genomes for antimicrobial targets," *Molecular Medicine Today*, vol. 6, 2000.
[3] J. Rosamond and A. Allsop, "Harnessing the power of the genome in the search for new antibiotics," *Science*, vol. 287, no. 5460, 2000, pp. 1973-1976.
[4] N. Jacq, C. Blanchet, C. Combet, E. Cornillot, L. Duret, K. Kurata, H. Nakamura, T. Silvestre, and V. Breton, "Grid as a bioinformatic tool," *Parallel Computing*, vol. 30, no. 9-10, 2004, pp. 999-1167.
[5] M. Cornell, I. Alam, D. Soanes, H. Wong, M. Rattray, S. Hubbard, N. J. Talbot, B. Lings, D. Hoyle, S. G. Oliver, and N. W. Paton, "e-Fungi: an e-science infrastructure for comparative functional genomics in fungal species," *Proc. 4th UK e-Science All Hands Meeting (AHM 2005)*, Nottingham, UK, September 20-22 2005.
[6] D. Sulakhe, A. Rodriguez, M. D'Souza, M. Wilde, V. Nefedova, I. Foster, and N. Maltsev, "GNARE: an environment for Grid-based high-throughput genome analysis," *Proc. 5th IEEE Int. Symp. Cluster Computing and Grid (CCGrid05)*, Cardiff, UK, May 9-12 2005.
[7] TIGR Grid Computing. Available: http://www.tigr.org/grid/
[8] NC BioGrid white paper. Available: http://www.ncbiogrid.org/about/NC-BioGrid_Flyer.pdf
[9] GPSA: Grid protein sequence analysis. Available: http://gpsa.ibcp.fr/
[10] EGEE. Available: http://public.eu-egee.org/
[11] COGs: phylogenetic classification of proteins encoded in complete genomes. Available: http://www.ncbi.nlm.nih.gov/COG/
[12] R. L. Tatusov, E. V. Koonin, and D. J. Lipman, "A genomic perspective on protein families," *Science*, vol. 278, 1997, pp. 631-637.
[13] 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.
[14] coliBASE. Available: http://colibase.bham.ac.uk/
[15] R. R. Chaudhuri, A. M. Khan, and M. J. Pallen, "*coli*BASE: an online database for Escherichia coli, Shigella and Salmonella comparative genomics," *Nucleic Acids Research*, vol. 32, no. Database, 2004, pp. D296-D299.
[16] R. Stevens, A. Robinson, and C. Goble, "myGrid: personalised bioinformatics on the information grid," *Bioinformatics*, vol. 19 Suppl 1, 2003, pp. i302-i304.

[17] R. Stevens, R. McEntire, C. A. Goble, M. Greenwood, J. Zhao, A. Wipat, and P. Li, "myGrid and the drug discovery process," *Drug Discovery Today: BIOSILICO*, vol. 2, no. 4, 2004, pp. 140-148.
[18] EGEE battles malaria with grid wisdom. Available: http://public.eu-egee.org/news/fullstory.php?news_id=53
[19] EMBL nucleotide sequence database. Available: http://www.ebi.ac.uk/embl/index.html
[20] BioJava. Available: http://www.biojava.org/
[21] Installing and using BioSQL. Available: http://www.biojava.org/tutorials/biosql.html
[22] A. Krishna, V. Tan, R. Lawley, S. Miles, and L. Moreau, "myGrid notification service," *Proc. UK e-Science All Hands Meeting*, Nottingham, September 2-4 2003, pp. 475-482.
[23] myGrid project. Available: http://www.mygrid.org.uk/
[24] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, 1990, pp. 403-410.
[25] NCBI BLAST. Available: http://www.ncbi.nlm.nih.gov/BLAST/
[26] 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.
[27] Ultra-fast alignment of large-scale DNA and protein sequences. Available: http://mummer.sourceforge.net/
[28] Globus Toolkit. Available: http://www.globus.org/toolkit/
[29] Condor. Available: http://www.cs.wisc.edu/condor/
[30] N1 Grid Engine 6. Available: http://www.sun.com/software/gridware/index.xml
[31] A. K. Bansal and T. E. Meyer, "Evolutionary analysis by whole-genome comparisons," *Journal of Bacteriology*, vol. 184, no. 8, 2002, pp. 2260-2272.
[32] Divide and conquer algorithm. Available: http://en.wikipedia.org/wiki/Divide_and_conquer_(computer_science)