Developing Grid-based Systems for Microbial Genome Comparisons: The Microbase Project

Anil Wipat, Yudong Sun, Matthew Pocock, Pete Lee, Paul Watson and Keith Flanagan

School of Computing Science, University of Newcastle upon Tyne

Abstract

Comparative analysis of genomes allows the rich source of biological genome sequence data to be most efficiently exploited. However, the rate at which microbial genomes are being sequenced is increasing rapidly. Soon the volume of data will put comparative analyses beyond the capability of the computing resources of most individual laboratories. Grid technology promises to provide a scalable solution for analysing genomic data. Microbase is a Grid-based system to support comparative genomics applications. The project aims to use Grid technology to provide an environment that is both scalable, provides access to pre-computed datasets and is also able to permit the execution of user defined, remotely conceived computation.

1. Background

1.1 Microbial Genome Sequencing and Comparison

Analysis of a microbial genome sequence can reveal novel information about the biology of a given organism by providing a catalogue of its genes and genome composition. To date, the sequences of 165 microbial genomes have been completed (http://www.tigr.org). Whilst single microbial sequences are themselves a gold mine of information, it is the comparative analysis of multiple genomes that allows this rich source of biological data to be most efficiently exploited. Comparative analysis of multiple genomes not only provides novel information about the physiology and evolution of microbial species but also provides a means of assigning putative function to genes of unknown function [1]. Phylogenetic analysis through comparative genomics is revealing new insights into the evolution of the microbial species, showing not only the similarities and difference between species but also helping to explain how such changes may have arisen [2, 3].

New strategies, tools and algorithms, that allow microbial genomes to be compared, are being developed in response to the increasing importance of comparative genomics. A number of general-purpose tools have been constructed that allow the visual comparative analysis of genome and genome segments, facilitating studies into genome synteny, gene order and protein similarity. Systems such as the comprehensive microbial resource (http://www.tigr.org), Alfresco [4], PipMaker [5], ACT [6] and Artemis [7] are examples of useful tools for this purpose.

Large-scale protein sequence comparison has also proved an effective strategy for extracting biological information from whole genome sequences. Protein sequence comparisons allow the relationships between homologous proteins to be classified in terms of paralogous proteins within genomes and orthologous proteins between genomes. They also permit studies into gene transfer events, and the identification of genes that have been acquired by lateral gene transfer, often termed Xenologs [8]. Several approaches have been proposed for performing pair-wise comparisons of whole genomes. Tausov and colleagues compared the protein sequences of complete genomes using the sequence similarity search algorithm BLAST [9, 10]. Their work resulted in the definition of clusters of orthologous proteins termed COGs, defined on the basis of consistent patterns in the graphs of best hits. Approximately 2790 clusters of orthologous groups were defined and a database was established to disseminate the results. The COG database is a particularly valuable resource for the functional characterisation of new proteins and for the phylogenetic annotation of gene sets.

As the frequency with which new genomes become available increases, automated techniques are becoming the first step for cross-species comparisons. Bansal and co-workers have described an automated framework

(GOLDIE) for performing pair-wise comparisons of microbial genomes [11]. The algorithm they developed uses a combination of BLAST and a Smith-Waterman alignment with a bi-partite graph matching procedure to identify clusters of orthologues and orthologous gene groups. The algorithm was successfully applied to the comparative analysis of 17 complete microbial genomes and used to provide evidence to suggest that gene group duplications are also involved in the evolution of genome functionality.

The alignment of the nucleotide sequence of whole genomes provides a method of high-resolution genome comparison that is complementary in approach to the pair-wise alignment of proteins from genomes. Systems based on the use of suffix trees, such as Mummer [12], can be used to rapidly align two large genomes, highlighting the exact differences between the genomes. The output from such comparisons includes matching segments of sequence termed MUMs (maximal unique matching subsequences), SNPs (one base mismatches), insertions (a sequence that occurs in one genome but not another), polymorphic regions (many mutations in short regions) and repeat sequences.

Tools that are effective for the comparison of microbial genomes are being continually refined. However, as more genome sequences are published, there is an increasing need to develop effective solutions for the storage and querying of the increasing number of genomes that will be available in the near future. Methods are also required that will allow genome comparison results to be maximally exploited by the biological community. To be most beneficial, the comparison data must be interpreted in the context of a range of other biologically relevant data (e.g. gene expression, protein function, protein interactions, metabolic pathways, virulence, environmental niche and taxonomy). The increasing volume of the genome databases raises a number of research problems that need to be addressed in the development of new genomic comparison applications. One such problem is the provision of adequate compute power to run even basic comparison algorithms on such a large number of genomes. In response to this problem a number of approaches have become available for genomic comparison using classical high performance parallel computing [13]. However, even with our current complement of complete microbial genomes, complete all-against-all genome comparisons are now impossible to perform locally using the resources available to the majority of researchers.

1.2 Grid Approaches to Sequence Comparisons

The advent of Grid technology promises to provide resources for computation, data integration and collaboration in a way that is not addressed in current distributed computing technologies. Grid computing has therefore been identified as having major potential benefits for bioinformatics, particularly in the area of genome analysis and comparative genomics [14, 15]. To date, however, the number of projects applying Grid technology to comparative genomics has been limited. There have been a few reports of Grids established to carry out large scale Blast analyses by large genome centres such as Argonne, TIGR and DOE Science Grid, but it is clear that additional technology is required before the routine use of Grid resources by the academic community becomes commonplace.

For example, PUMA [16] developed by the Computational Biology Group of Argonne National Laboratory is an integrated computational framework for comparative analysis of cellular metabolism. It is designed to support high-throughput analysis of genomes using Grid technology for comparative and evolutionary analysis of metabolic processes on various levels of biological organization in the context of phenotypic and taxonomic information derived from authoritative sources.

TIGR provides an in-house repository of protein and nucleotide data made available by major genome data repositories such as GenBank, Protein Information Resource (PIR), and SwissProt. For creating a custom non-redundant protein databases for annotation, TIGR performs an all-vs.-all search on all the proteins from these sources to create clusters of similar proteins. To perform the all-vs.-all search TIGR partitions the data set into multiple subsets and runs NCBI BLAST searches in parallel on the Grid [17]. The Grid is first used to pre-filter the data subsets to identify proteins in the database with some degree of similarity to query proteins. A more exhaustive protein search is then performed on the pre-filtered data using NCBI Blast to identify protein similarity.

GADU (Genome Analysis and Databases Update tool) [18] is a collaborative project between Globus project and the Argonne bioinformatics group. It has developed an automated, high-performance, scalable computational pipeline for data acquisition and analysis of the newly sequenced genomes with

DOE Science Grid backend [19]. GADU allows efficient automation of major steps of genome analysis including data acquisition, data analysis by variety of tools and algorithms, as well as data collection, storage and annotation.

2. Project Aims

One aim of the Microbase project is to develop a system for carrying out genome comparisons that is scalable to cope with the influx of new genomes by harnessing Grid based resources. The project is also researching Grid-based methods that allow genome comparison results to be interpreted in the context of a range of other biologically relevant data (e.g. gene expression, protein function, protein interactions, metabolic pathways, virulence, environmental niche and taxonomy). Genomic comparison datasets are too large to be analysed at the client side. The Microbase project is developing a scalable Grid based system that is able to provide a pre-computed genome comparison dataset and provide an environment to support user defined, remotely conceived, computationally intensive algorithms, operating over this data. The major aim of Microbase is to research and develop a Grid environment to which user defined computations may be submitted to operate over pre-computed genome comparison datasets, integrating data from other Grid enabled databases as required. It is envisaged that the user-submitted computations may remain in place on the server side to be re-executed as new genomes appear. The project aims to exploit Grid specific features, such as provenance tracking, user notification, resource discovery and workflow enactment. Open source components to facilitate the use of these features are now starting to become available as myGrid [20] and other Grid middleware projects mature.

3. Microbase-*Lite*

The first deliverable of the project is a prototype system, Microbase-*Lite*, that acts as a demonstrator, establishes a data repository and drives the requirements gathering process for subsequent research into the mature Microbase system. Microbase-*Lite* has been developed using mostly conventional technology, and has been designed to provide a web service based repository of genomes, genomic comparison results and genome comparison algorithms to support bioinformaticians in the development of new genomic comparison tools. The preliminary architecture of the Microbase-*Lite* system is shown in Figure 1.

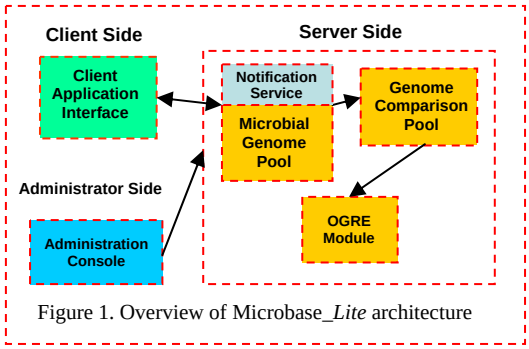

Figure 1. Overview of Microbase_*Lite* architecture

The Microbase-*Lite* system comprises a number of distinct components that interoperate through web service interfaces. The components which currently make up Microbase-*Lite* are the microbial genome pool, the genome comparison pool, the notification service and the OGRE module. In addition a simple client application has been constructed to act as a focus for the user and to test the service interfaces. Each of these components is discussed in more detail below.

3.1 The Microbial Genome Pool

The first of these components to be designed and implemented is a standalone Microbial Genome Pool (Fig. 2). It provides a local source of complete microbial genome sequences (165 sequences in June 2004) and has a web service interface that provides access to complete or subsections of microbial genome sequences and their features.

The genome pool is automatically kept up to date using functionality provided by the myGrid notification service (see 3.4). A notification event is sent to the genome pool when new genomes arrive in the EMBL database and the new genome sequence is retrieved. External service based components may register with the genome pool to receive notification and details of the arrival of the new genome into the pool. The implementation of the genome pool has been carried out using Biojava (http://www.biojava.org). EMBL records for complete genomes are parsed and stored in a relational database based around the PostgreSQL implementation of the BioSQL schema. The service interface allows users to query the nucleotides, proteins and other features of the genome sequences stored in the database as well as register for the notification service through the Internet.

The microbial genome pool is available for use by external users that require notified,

computational access to microbial genome sequences. A description of the functionality and service interface for the microbial genome pool can be found at http://vindaloo.ncl.ac.uk:8090/genomepool/index.html.

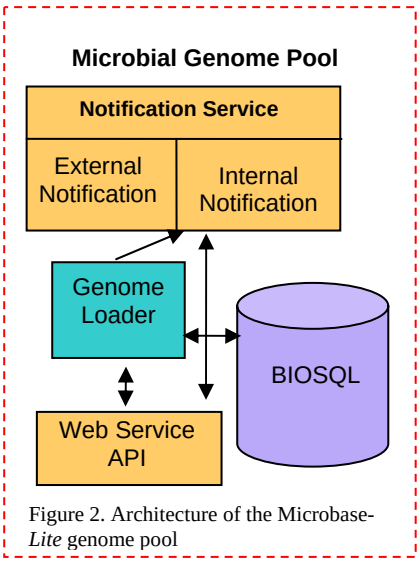

Figure 2. Architecture of the Microbase-*Lite* genome pool

3.2 The Genome Comparison Pool

New genomes are compared to the existing genomes in the genome pool database at the nucleotide and protein sequence level by a genome comparison pool component (Fig.3). The comparison pool employs a variety of algorithms including BLAST and suffix tree based sequence comparisons such as Mummer. A task scheduler decides which programs should be run on what data and in which order, and then manages the submission of compute-intensive tasks to a parallel cluster using N1 Grid Engine (formally Sun Grid Engine) [21] to manage the jobs. Comparison results are stored in a relational database using MySQL. In addition, the comparison pool includes a number of algorithms which are employed to provide orthologue and paralogue determination and protein family classifications.

3.2.1 Task Scheduler

Microbase-*Lite* performs microbial genome comparisons at the nucleotide and protein levels using different tools. The tools include blastn (nucleotide-nucleotide BLAST), blastp (protein-protein BLAST), Mummer, Promer (Protein Mummer), Mspcrunch [22] and Ssearch [23]. To date, 165 microbial genomes have been loaded into the genome pool. The all-to-all comparison with the six tools needs to perform as many as 163,350 separate computing jobs. This number is soaring as the genome pool database is regularly updated by adding new genomes into it, and all genomes must be compared against all others. The individual sequence comparisons are data- and cpu-intensive operations. For example, the protein comparison between two bacteria, *bacillus cereus* (AE016877) and *bacillus anthracis* (AE016879), using blastp requires two input files with the size of 1.5 MB each and produces an output file in the size of 95 MB. The comparison takes 12 minutes on a 2.8 MHz Intel Xero processor. Therefore, the cumulative time to run all-to-all comparisons for a large number of genomes is considerable.

To deal with the large amount of data-intensive computing, a task scheduler has been developed as a component of the comparison pool to farm the jobs of genome comparisons to a cluster of workstations and to manage the parallel execution of the jobs. The task scheduler implements the scheduling function by calling the N1 Grid Engine. Like other job-management Grid middleware, N1 Grid Engine accepts jobs submitted by users and schedules the jobs to be run on appropriate hosts in the Grid according to the resource management policies.

To run a genome comparison, the task scheduler creates a thread. The thread runs autonomously to retrieve the nucleotide or protein sequences of a pair of genomes from the genome pool database as the input data, and constructs a comparison job by feeding the input data to a comparison tool such as blastn or blastp. Then, the thread calls a N1 Grid Engine script that in turn submits the job to a processing node in the cluster for execution. The thread periodically checks the status of the job. When the job has completed, the thread performs an analysis on the output data. A parser is implemented for the comparison result generated by a comparison tool. The thread calls the corresponding parser to extract the required data from the raw output and stores the data into the comparison pool database. With the multithread execution, the task scheduler can manage numerous comparison jobs in parallel. After all threads have completed their comparison jobs, the task scheduler terminates the comparison procedure.

As previously indicated, a large number of jobs will be created to run the genome comparisons. To prevent the active jobs from overwhelming the system resources, the task scheduler uses a threshold policy to control the number of jobs being created. The threshold is

set to the maximum number of processors that can be used to run the jobs. The task scheduler manages the rate of job creation not exceeding the threshold to avoid system congestion.

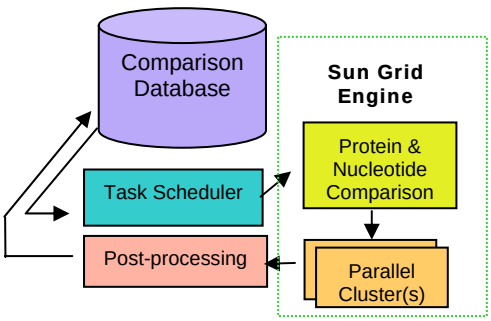

Figure 3. Architecture of the Microbase-*Lite* comparison pool.

3.3 The Notification Service

A notification service is implemented based on the myGrid notification system [24] to notify the registered clients and coordinate the processing of new genomes by the Microbase-*Lite* system as the new genomes arrive in the genome pool. The myGrid notification service is a general-purpose Web Service for event notification based on JMS (Java Message Service), which supports topic-based publish-subscribe messaging, push/pull models and asynchronous delivery. Microbase-*Lite* uses the notification service to inform registered clients about the arrival of new genomes and triggers the task scheduler in the comparison pool to run genome comparisons for the new genomes.

3.4 OGRE

A further novel feature provided by the Microbase-*Lite* 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 [6, 25]. OGRE intends to develop a formally defined set of terms relating to genome rearrangement in the form of an 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 algorithms, and an object oriented data model for the comparison and analysis of genome sequences. OGRE is a sister project to Microbase-*Lite*, 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 active development, and there is a working prototype capable of detecting some simple features.

3.5 Client Application

A graphical genome comparison viewer in the form of a Java application is under development and will act as a client to allow a biologist to access the Microbase-*Lite* system. This client will be made available for use by academic partners and will provide a means by which a user can remotely browse genomes, view genome and gene comparison data and formulate their own queries for execution within the system. The graphical viewer consists of two components: query builder and genome browser. The query builder shown in the lower part of Figure 4 allows users to specify their requirements for searching the OGRE dataset, such as source sequence, target sequence, a specific comparison, and the scope of comparison results. The genome browser shown in the upper part of Figure 4 displays the result retrieved from the dataset in response to the user's query. A graphical representation is used to show the result of query to user.

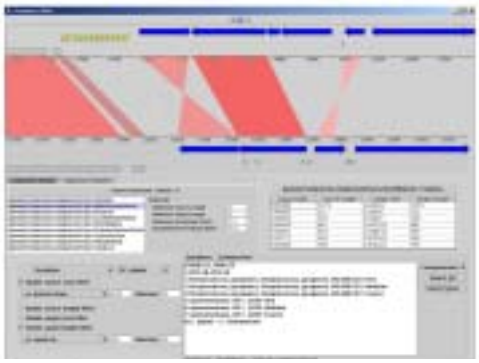

Figure 4. Screenshot of the graphical viewer for the genomic comparison explorer application

3.6 Results: Performance and Scalability

With the support of the task scheduler, the comparison pool can achieve scalability in executing data-intensive comparison jobs. Figure 5 shows the performance of all-to-all comparisons among 10 bacterial genomes using the six tools indicated in 3.2.1. The nucleotide sequences of these genomes have a length in the range of 0.5M to 8M base pairs. Seven of them are in the length of 4M to 5M base pairs. The comparisons are performed on a cluster of

workstations, and use between one and 30 processors.

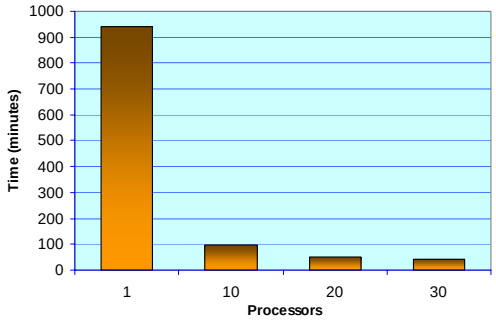

Figure 5. Execution time of all-to-all genome comparisons

As Figure 5 shows, the all-to-all comparisons of 10 genomes can be accelerated when more processors are employed to run the comparisons in parallel. We plan to incorporate more processing nodes into our computing system and test the scalability of the comparison pool on the enlarged system base.

4. The Vision for the Full Microbase System

Microbase-*Lite* has established the base level dataset, established a demonstrator system and begun to explore and test scalability issues. The results obtained must be evaluated and the experiences gained used to shape the development of the full Microbase system.

However, two major requirements have not been addressed by the Microbase-*Lite* system. Firstly, the ability of the system to scale seamlessly to meet the demands of an almost exponential influx of new genomes still remains to be realised. Secondly, an additional complete system is required to allow remotely conceived code or workflows to be submitted to and enacted within the Microbase system.

Thus, the major focus of the Microbase project will now be on the research required to add the functionality to meet these requirements. One avenue of research we will pursue is to harness remote computing clusters from Microbase on demand, for example, by using functionality provided by the N1 Grid Engine.

Research is also required to establish a suitable framework within Microbase for the execution of user analysis. One promising approach to meeting this requirement is through development of a system that allows the creation, submission and execution of user-defined workflows and services. These may be executed with direct access to many of the computational and database resources used by Microbase itself. Grid based workflows for bioinformatics are the focus of the Taverna project [26] and we are currently investigating mechanisms for the submission and enactment of Taverna workflows within the Microbase system.

5. Summary

The Microbase project is building a Grid based environment to support both bioinformaticians and biologists wishing to perform genome comparisons and associated analyses. Our initial demonstrator system, Microbase-*Lite*, is designed to provide service-based access to pre-computed genomic comparison data. The API for the completed Microbase-*Lite* systems will be available for use at the time of the publication of this paper. Overall, the Microbase project aims to provide systems that will support the biological and bioinformatics community, by allowing them to perform their own microbial genomic analysis in a remote high performance environment. Such systems will not only store and return their experimental results but will ultimately provide seamless access to additional analytical services and data sources over the Grid.

Acknowledgments

Microbase 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.

References

1. I. Iliopoulos, S. Tsoka, M. Andrade, et al., *Evaluation of annotation strategies using an entire genome sequence.* Bioinformatics, 2003. **19**(6): p. 717-26.
2. B. Williams, R. Hirt, J. Lucocq, et al., *A mitochondrial remnant in the microsporidian Trachipleistophora hominis.* Nature, 2002. **418**(6900): p. 865-9.
3. R. Hirt, S. Muller, T. Embley, et al., *The diversity and evolution of thioredoxin reductase: new perspectives.* Trends Parasitol, 2002. **18**(7): p. 302-8.
4. N. Jareborg and R. Durbin, *Alfresco--a workbench for comparative genomic*

sequence analysis. Genome Res, 2000. **10**(8): p. 1148-57.
5. L. Elnitski, C. Riemer, H. Petrykowska, et al., *PipTools: a computational toolkit to annotate and analyze pairwise comparisons of genomic sequences.* Genomics, 2002. **80**(6): p. 681-90.
6. ACT (Artemis Comparison Tool), http://www.sanger.ac.uk/Software/ACT/
7. M. Berriman and K. Rutherford, *Viewing and annotating sequence data with Artemis.* Brief Bioinform, 2003. **4**(2): p. 124-32.
8. M. Omelchenko, K. Makarova, Y. Wolf, et al., *Evolution of mosaic operons by horizontal gene transfer and gene displacement in situ.* Genome Biol, 2003. **4**(9): p. R55.
9. R. Tatusov, N. Fedorova, J. Jackson, et al., *The COG database: an updated version includes eukaryotes.* BMC Bioinformatics, 2003. **4**(1): p. 41.
10. S. Altschul, T. Madden, A. Schaffer, et al., *Gapped BLAST and PSI-BLAST: a new generation of protein database search programs.* Nucleic Acids Res, 1997. **25**(17): p. 3389-402.
11. A. Bansal and T. Meyer, *Evolutionary analysis by whole-genome comparisons.* J Bacteriol, 2002. **184**(8): p. 2260-72.
12. S. Kurtz, A. Phillippy, A. Delcher, et al., *Versatile and open software for comparing large genomes.* Genome Biol, 2004. **5**(2): p. R12.
13. N. Almeida, C. Alves, E. Caceres, et al. *Comparison of genomes using high-performance parallel computing*. in *Comparison of Genomes Using High-Performance Parallel Computing*. 2003. São Paulo, SP - Brazil: IEEE.
14. A. Krishnan, *A survey of life sciences applications on the Grid.* New Generation Computing, 2004. **22**(2): p. 111-125.
15. A. Konagaya, F. Konishi, M. Hatakeyama, et al., *The superstructure toward open bioinformatics grid.* New Generation Computing, 2004. **22**(2): p. 167-176.
16. M. D'Souza, J. Huan, S. Sutton, et al., *PUMA2 - An environment for comparative analysis of metabolic subsystems and automated reconstruction of metabolism of microbial consortia and individual Organisms from sequence data*. 1999
17. TIGR Grid Computing, http://www.tigr.org/grid/
18. A. Rodriguez, D. Sulakhe, E. Marland, et al., *GADU - Genome analysis and database update pipeline*. 2003
19. DOE ScienceGrid: summary of progress, Feb. 2002 to Feb. 2003, http://www.doesciencegrid.org/Grid/management/
20. R. Stevens, A. Robinson, and C. Goble, *myGrid: personalised bioinformatics on the information grid.* Bioinformatics, 2003. **19 Suppl 1**: p. 302-4.
21. N1 Grid Engine 6 user's guide, http://docs.sun.com/db/doc/817-6117
22. E. Sonnhammer and R. Durbin, *A workbench for large scale sequence homology analysis.* Comput. Applic. Biosci., 1994. **10**: p. 301-307.
23. T. Smith and M. Waterman, *Identification of common molecular subsequences.* J. Mol. Biol., 1981. **147**: p. 195-197.
24. A. Krishna, V. Tan, R. Lawley, et al. *myGrid Notification Service*. in *UK e-Science All Hands Meeting*. 2003.
25. J. Yang, J. Wang, Z. Yao, et al., *GenomeComp: a visualization tool for microbial genome comparison.* Journal of Microbiological Methods, 2003. **54**: p. 423-426.
26. T. Oinn, M. Addis, J. Ferris, et al., *Taverna: a tool for the composition and enactment of bioinformatics workflows.* Bioinformatics, 2004.