1. General management

We held two workshops to report on research progress and promote close cooperation. We participated in the International Cancer Genome Consortium (ICGC) 6th Scientific Workshop held in France on March 21–22, and gathered information regarding large-scale data analysis methods and the status of research on the use of next-generation sequencers in cancer genome analysis. We also delivered a talk on large-scale life data analysis at the Biophysical Society of Japan's "High Performance Computational Approaches to Biological Functions" symposium held on September 16, 2011.

In addition to the above, we also held periodical meetings among research program participants to coordinate research activities.

2. Development of data processing system for next-generation sequencer data analysis

We developed GHOSTX, a sequence homology search tool, and parallelized GHOSTX with an MPI/Open MP hybrid to create GHOST-MP. Using the K computer, we optimized the GHOST-MP code and database to improve execution speed and parallel performance within and between nodes (Figure 1).

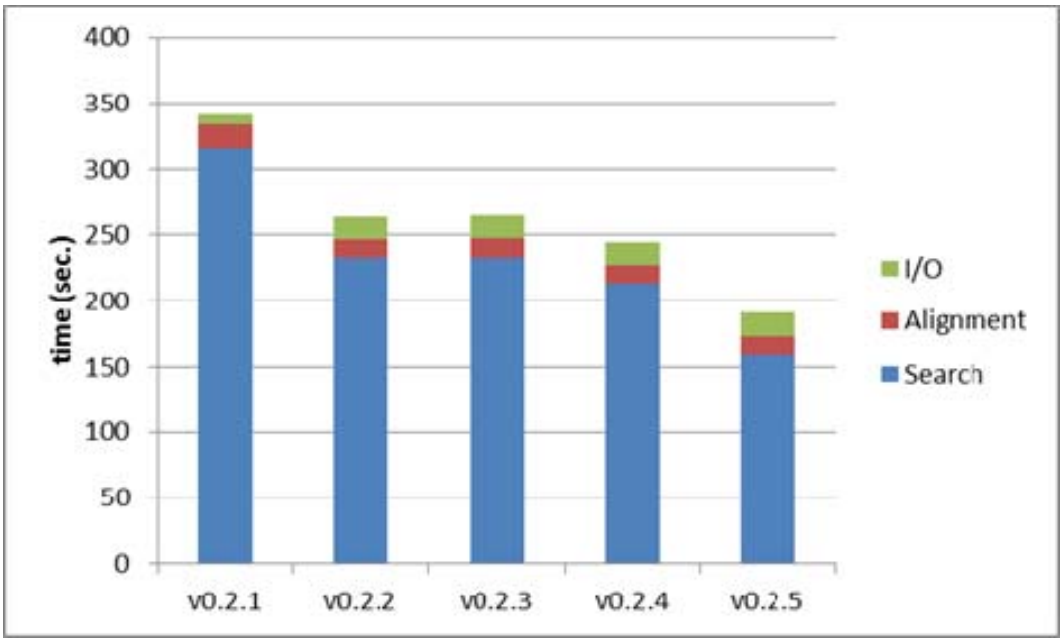

Figure 1. Execution times when applying GHOST-MP to metagenomic analysis

The horizontal axis shows the GHOST-MP version, and vertical axis, execution times. "I/O", "Alignment" and "Search" represent execution times respectively for file I/O, homologous sequence alignment, and homologous sequence search. Execution speed was improved with Ver.0.2.5 being 1.7 times the speed of Ver.0.2.1.

For this research project, we used a suffix array for both query and database sequences to enable GHOSTX to perform sensitive, high-speed searches. We compared GHOSTX with the BLAST and BLAT sequence homology analysis tools using actual data (fragments of 60–75 bases) from soil microorganism metagenomes and simulated data for fragments of about 500 and 1000 bases. To assess sensitivity, we compared agreement of results with SSEARCH results, assuming the latter to be correct since the SSEARCH program uses the Smith-Waterman algorithm to make very precise calculations of optimum alignments. When we ran GHOSTX using parameters prioritized for speed, it proved to be faster than both BLAST and BLAT, completing searches at about 100 times

the speed of BLAST. Sensitivity was lower than BLAST, but higher than BLAT. When using parameters prioritized for sensitivity, search speed was slower than BLAT, but about 20 times faster than BLAST, while delivering similar sensitivity (Figure 2).

In short, our GHOSTX tool enables homology analysis at the speed and sensitivity required for metagenomic analysis.

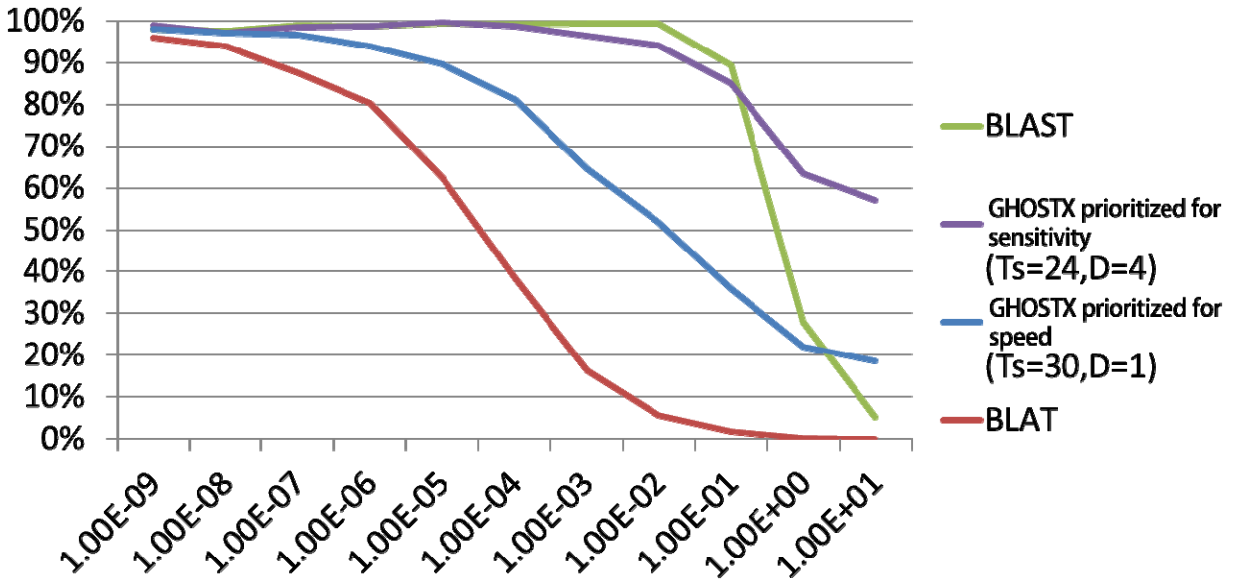

Figure 2. Comparison of the sensitivity of GHOSTX, BLAST and BLAT
The degree of agreement with the results of SSEARCH is shown for each tool.
The vertical axis shows agreement rate with SSEARCH results, and horizontal axis, E-values.
The agreement rates are calculated for each E-values.

3. Development of methods for predicting RNA-RNA interaction and comprehensive analysis of RNA data

Since any analysis of RNA sequence data needs to consider secondary structure in order to produce accurate results, comprehensive analysis of RNA data also requires consideration of secondary structure. We accordingly developed a method that takes secondary structure into account, and carried out molecular simulation to analyze the tertiary structure of RNA molecules.

To develop methods for analyzing RNA secondary structure, we developed Raccess, a software package that computes the accessibility of all the segments of a fixed length for a given RNA sequence, and IPknot, a software package for predicting RNA secondary structures with pseudoknots based on maximizing the expected accuracy of a predicted structure. We have published papers in international journals on both tools.

For molecular simulation, we also looked into developing a method that packages coarse-grained modeling of RNA molecules with parameters for predicting base-pairing probability and other aspects of RNA secondary structure.

For tertiary structure analysis, we developed RASSIE, a fragment assembly software package for accurately predicting RNA tertiary structures by using known secondary structure information. We subsequently published details of RASSIE.

We also developed a long string NGS simulation data assembly tool for RNA-RNA interaction evaluation to consider the effectiveness of applying RNA-RNA interaction prediction techniques to comprehensive analysis of RNA data.

4. Development of large-scale biomolecular network analysis techniques

We devised a seed network method as a large-scale biomolecular network inference method that incorporates biological knowledge, and developed a software prototype for inferring intermolecular control networks. We then extended this prototype on the K computer to enable large-scale parallel execution. We named this tool BENIGN (Biologically Extensible Network Inference Software for Gene Expression Analysis).

Table 1 shows BENIGN's parallel execution performance on the K computer. For data, we used changes in gene expression profile during the differentiation of mouse mesenchymal stem cells into adipocytes. The gene set that we focused on was 113 genes known to be involved in adipocyte differentiation, and genes encoding 833 transcription factors that showed changes in expression during differentiation, making for a total of 946 genes.

Table 1. BENIGN's parallel execution performance

Calculated nodes	Execution time (seconds)	Speed improvement	Parallel efficiency
6,114	2683	1.00	
12,228	1386	1.94	0.97

In Table 1, execution time for 6,114 nodes is used as a base for measuring speed improvement and parallel efficiency by strong scaling. As the table shows, excellent parallel efficiency of 97% was achieved with the execution of 12,228 nodes.

We were also able to confirm the validity of the results of biomolecular network analysis for the process of adipocyte differentiation using BENIGN.

Going forward, we plan to enhance BENIGN's dynamic load distribution function and other components, and make the most of K's immense processing power by conducting network inference on a whole gene set that includes other genes in addition to transcription factor genes. We have also been supplied with an assortment of RNA-Seq gene expression data for mouse adipocyte tissue by research colleague Professor Teruo Kawada of Kyoto University's Graduate School of Agriculture, and we plan to use it to attempt the inference of large-scale biomolecular networks for total RNA, including miRNA and other non-coding RNA.

5. Metagenomic analysis and comparative genome analysis research

Table 1. Comparison of maximum likelihood programs for phylogenetic tree inference

Package name	Methods	Parallelization
fastDNAml	ML	○
MEGA	MP, ML	
MOlPHY	ML	
MrBayes	BI	○
PAML	ML, BI	
PHYLIP	MP, ML	○
RAxML	MP, ML	◎
TREE-PUZZLE	ML	○

ML stands for maximum likelihood, MP for maximum parsimony, and BI for Bayesian inference.

Phylogenetic tree inference involves the multiple alignment processing of base sequences obtained from genomes to infer phylogenetic trees. In conjunction with advances in computational performance in recent years, the phylogenetic tree inference method that we used for this research project—the maximum likelihood method—is being used increasingly in life science research fields, and it has won broad acceptance. The maximum likelihood method runs comparisons among multiple candidate phylogenetic trees to select the phylogenetic tree shape that best matches the given sequence data. Inferring the best tree shape requires the comprehensive searching of all candidate phylogenetic trees, but since the number of candidate tree shapes grows exponentially to the order of $O(2^N N!)$ where N is the number of sequence fragments, the comprehensive searching of whole candidate tree shapes for more than a certain number of sequences is theoretically impossible. As such, search space needs to be narrowed down when implementing the program in order to enable maximum likelihood inference. Since search methods differ according to program, implementing programs on the K computer requires different parallel optimization for each program. We accordingly prepared for this research project by first investigating programs for their compatibility with the K computer's parallel architecture (Table 1). Our investigation found that RAxML, a program developed by A. Stamatakis and others at Germany's Ludwig-Maximilians University, is the most suitable program, so we used the RAxML code as a base for developing our software.

Since it uses MPI, the RAxML program in principle meets the parallel programming requirements of the K computer. However, we found that RAxML could not be used as it is on the K computer because it also incorporates other parallelization methods such as p-Thread in parts. We accordingly implemented the program with parallelization methods that could be used on the K computer. We compared the parallel efficiency of K's architecture with a PC cluster, and found that the RAxML program is in principle compatible with the K computer's architecture.