**Project Introduction**

**Objectives**
To provide a programming toolkit to facilitate the development and deployment of large-scale bioinformatics applications on a Grid.

**Project Investigator / Manager**
A/Prof. Teo Yong Meng
National University of Singapore
teoym@comp.nus.edu.sg

**Period of Project**
Jun 2000 – Jun 2003

**Website**
www.comp.nus.edu.sg/~teoym/atsuma.htm

**Abstract**
GLAD (Grid Life sciences Applications Developer) is a grid-based bioinformatics applications development workbench. It was developed using the ALiCE grid system, a Java-based grid middleware that exploits task-based parallelism.

**Project Details**

**Description**
Increasingly Grid Computing is used to solve compute-intensive bioinformatics problems with large databases by distributing the computation across geographically distributed resources. However, knowledge and understanding of parallel computing algorithms and techniques and the need to handle geographically distributed sequence databases of varying formats increase the complexity of developing bioinformatics applications for different classes of problems.

GLAD, developed by extending the ALiCE API, aims to reduce the development effort of distributed bioinformatics applications by providing a library of commonly used components such as phylogeny module and similarity module. ALiCE is a Java-based Grid system that we have developed. The GLAD architecture is divided into four main layers. The execution layer consists of a parallelization engine that automatically exploits applications with regular and semi-regular parallelism patterns, data access and parser engines manage bioinformatics data, and a statistics generator produces commonly-used performance measures. To reduce development time, a tool library provides a set of commonly-used bioinformatics algorithms. Application development is supported by using the problem description template, a Java-based programming template for users to model bioinformatics applications.

GLAD Architecture

The development framework allows a programmer to customize the application’s user interface by selecting from a library of standardized visualization components such as the graphical illustration of DNA and protein sequences, textual display of results, or the implementation of his/her own visualization. Benchmark applications developed include distributed sequence comparison and distributed progressive multiple sequence alignment.

**Collaborating Organization:**
Bioinformatics Institute

**Reference**