

**Keasar** - 283 models for 58 3D targets

## **MESHI – a new object oriented package for molecular simulations**

N. Kalisman, A. Levi and C. Keasar

*Department of Computer Science, Ben-Gurion University, Israel*

keasar@cs.bgu.ac.il

MESHI is a novel software package that handles many aspects of molecular simulations. The motivation behind MESHI is twofold (1) to shorten the delay between the emergence of a novel idea (say, while one is doing the dishes) and the testing of its programmed manifestation. (2) to lower the “activation barrier” of the code, i.e. reduce the time it takes a new developer to start writing new modules. In order to achieve these goals, MESHI adheres to a strict Object Oriented Design (OOD) and emphasizes clear code, even at the expense of some computational efficiency. In CASP6 we tried to demonstrate that while still in a stage of development, MESHI has already crossed the critical point where useful molecular modeling is possible.

In practice, strict OOD implies that every aspect of molecular modeling is represented by a class. Thus, MESHI is equipped with classes for molecular elements (e.g. atoms and residues), geometric properties (e.g. distances and angles), energy-terms (e.g. hydrogen-bonds), optimization-algorithms (e.g. steepest-descent and LBFGS) and quite a few auxiliary classes (e.g. PDB formatted line). These classes serve as handy building blocks to MESHI applications like BEAUTIFY (the program we used for CASP6).

MESHI is written solely in Java, which is not the obvious language of choice for a computationally intensive program. Its interpreted nature is inherently slower than native binary code. Our experience is that java code is about two times slower than equivalent C/Fortran code. We believe however, that the most precious resource is the developer's time, as Moore's law does not apply to it. The strict object oriented nature of Java forces a highly modular program structure and helps in optimizing human effort. Further, Java's garbage-collection utility seems to remove a large family of bugs from our way. In practice, the performance loss of Java is much lower than twofold. By profiling one can easily identify the (typically few) bottlenecks where the program spends most of its time. These parts of the program may be written with emphasis on performance and/or compiled to a binary module.

Due of its low “activation barrier”, MESHI is handy as an educational tool. In the last three years, students at the bioinformatics track of BGU did interesting and substantial projects within MESHI. The projects were defined in terms of interfaces and the students could focus in their specific tasks without diving into the code too deeply.

MESHI is free for academic use, and is available at:

<http://www.cs.bgu.ac.il/~keasar/meshi>