MIST: Molecular Integration Simulation Toolkit

Iain Bethune[1]*, Elena Breitmoser[1], Gianpaolo Gobbo[2], Charles Matthews[2] and Ben Leimkuhler[2]

*ibethune@epcc.ed.ac.uk [1]EPCC, University of Edinburgh [2]School of Mathematics, University of Edinburgh

Overview

One of the current Grand Challenge problems in Biochemistry is being able to understand the properties of complex macromolecular systems by effective sampling of their conformational space using Molecular Dynamics. Brute-force sampling using very long MD runs is inadequate since even with optimised software and specialised parallel hardware (e.g Anton, MDGRAPE) it is only possible to sample up to milliseconds of dynamics. This may not be enough to observe transitions which occur on biological timescales, and certainly is not enough to obtain robust statistics. Several well-known approaches exist to enable MD simulations to escape from local energy minima and explore the phase space of a system such as metadynamics, parallel tempering, steered MD and replica exchange. However, these methods have limitations such as the inability to recover an equilibrium (Boltzmann-weighted) sampling of the space, or require *a priori* knowledge of a system, such as suitable collective variables to distinguish the important conformations of a system.

The NSF / EPSRC [EP/K039512/1] funded ExTASY project (Extensible Tools for Advanced Sampling and analYsis) proposed a three-pronged solution to this challenge – coupling together the efficient execution of large ensembles of independent or loosely coupled MD trajectories, advanced analysis tools to efficiently sample the entire phase space, and the use of novel integration methods that allow extremely long time steps and biased sampling using collective variables.

The topic of this poster, the Molecular Integration Simulation Toolkit (MIST), addresses third aspect of the ExTASY approach to sampling. In order for new integration algorithms to be widely adopted by the biomolecular simulation community, their effectiveness must be demonstrated on systems of biological relevance. Typically, integrator development is carried out within simplified MD codes, which lack the efficient force evaluation and parallelisation approaches available in 'production' MD codes such as GROMACS, NAMD and LAMMPS, which have large existing user bases. The excellent performance and scalability of these codes comes at a cost of software complexity, and thus there is a barrier to the implementation of new algorithms, their testing at scale, and their eventual adoption by the wider user community.

MIST overcomes this barrier by providing two things - a simple API for integrator developers which hides the complexities of real MD codes, enabling rapid development of new integration algorithms at a higher level of abstraction; and a plug-in interface, enabling the new integrators to be deployed directly in several MD codes, taking advantage of their highly efficient force evaluation code, file format support, and supporting tool chains.

ExTASY
Extensible Tools for Advanced Sampling and Analysis

epcc | EPSRC
Engineering and Physical Sciences Research Council

THE UNIVERSITY OF EDINBURGH

MIST Library Architecture

MIST is a C++ library, and provides two key abstractions, each implemented as an abstract class:

- The *System* consists of a set of particles, with various properties: position, velocity, mass, force on the particle, and species. A simple API is provided to get and set the properties of each particle. In addition, accessors are provided for global quantities such as the potential energy, and a single function call updates the forces on the particles given the current state of the system. Subclasses are provided for each supported MD code, which implement the System API using the data structures present in that code.

- An *Integrator* provides only a single method, which integrates the system from time `t` to `t+dt`. Integrators are implemented using only the System API, completely independent of any particular MD code. Several examples are provided, including a Verlet integrator, 4th and 8th order Yoshida symplectic integrators, Langevin dynamics using the BAOAB scheme [1], and a Continuous Tempering integrator [2].

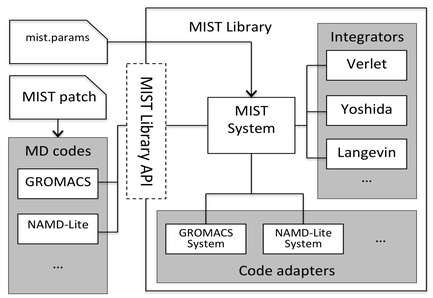

Figure 1: MIST Library Architecture.

MIST also provides a C API to enable integration with 'host' MD codes. MIST API calls are inserted in place of the usual MD step via source-code patches, which are provided with MIST and applied during the build process.

To use MIST requires setting a single configuration flag in the host MD code, then further configuration of MIST is via a separate `mist.params` file, for example selecting an integrator and setting any additional integrator parameters. All configuration of the force-field, simulation parameters and I/O is done as normal in the MD code's input files.

Performance

The MIST library is designed to provide a high-level abstraction of an MD code, to enable rapid development of new integrators. However, this should not have a significant effect on application performance.

To measure the impact of using MIST, we simulated a 50Å^3 box of water molecules using a flexible TIP3P model from the CHARMM forcefield for a total of over 11,000 atoms. Verlet integration and an NVE ensemble with a time step of 1 fs, were used. The input files for this system can be found in the `examples` directory of the MIST distribution. Both NAMD-Lite version 2.0.3 and Gromacs version 5.0.2 were tested, in three configurations:

- NO_MIST: Code built normally without MIST library
- MIST_OFF: Code linked with MIST library, using native Verlet integrator
- MIST_ON: Code linked with MIST library, using MIST Verlet integrator

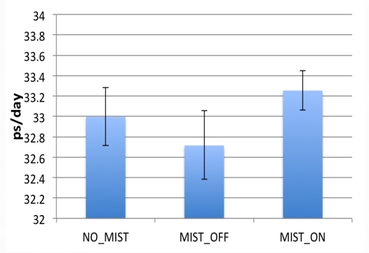

Figure 2: Performance of NAMD-Lite with/without MIST

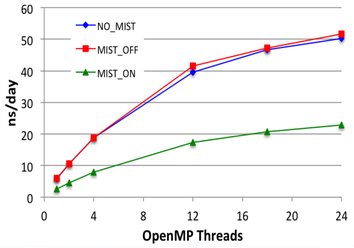

Figure 3: GROMACS scaling with/without MIST

NAMD-Lite performance tests were carried out on a MacBook Pro with a quad-core Intel Haswell 2 GHz processor. As shown in Figure 2, linking with MIST has negligible impact on the performance of NAMD-Lite. Indeed, the MIST Verlet integrator appears to be around 2% more efficient than the native implementation in NAMD-Lite. GROMACS was tested on ARCHER, a Cray XC30 with 2 12-core Intel Ivy Bridge 2.7 GHz processors per node. Figure 3 demonstrates that MIST can make use of OpenMP threading to increase performance, but there is currently a significant overhead of about 2.5 times due to the use of MIST for the integration step. Nevertheless, GROMACS is still over 100 times faster than NAMD-Lite, due to its highly optimised force evaluation implementation, so can be used for production-scale simulations in combination with MIST.

Download MIST

MIST is freely available (BSD License) from :

http://www.extasy-project.org/mist

The following features are available in the MIST 1.0 release:

- Plug-ins for GROMACS and NAMD-Lite (OpenMP parallelism only)
- Verlet, Yoshida (4th / 8th order), Langevin, Continuous Tempering Integrators
- Access to individual components of the potential energy / per-particle forces

In the next version of MIST (mid-2015), we plan to include the following:

- Plug-in for AMBER
- MPI parallel execution
- Support for constraints (SHAKE, SETTLE ...)

Please download and experiment with the code - we welcome feedback, contributions and suggestions!

[1] B. Leimkuhler and C. Matthews. (2012) "Rational Construction of Stochastic Numerical Methods for Molecular Sampling," Applied Mathematics Research Express, Vol. 2012

[2] See poster by G. Gobbo and B. Leimkuhler