Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Molecular Graphics and Modelling 22 (2004) 377–395

Journal of Molecular Graphics and Modelling

www.elsevier.com/locate/JMGM

MMTSB Tool Set: enhanced sampling and multiscale modeling methods for applications in structural biology

Michael Feig, John Karanicolas, Charles L. Brooks III*

Department of Molecular Biology, TPC6, The Scripps Research Institute, 10550 North Torrey Pines Road, La Jolla, CA 92037, USA

Abstract

We describe the Multiscale Modeling Tools for Structural Biology (MMTSB) Tool Set (http://mmtsb.scripps.edu/software/mmtsbToolSet.html), which is a novel set of utilities and programming libraries that provide new enhanced sampling and multiscale modeling techniques for the simulation of proteins and nucleic acids. The tool set interfaces with the existing molecular modeling packages CHARMM and Amber for classical all-atom simulations, and with MONSSTER for lattice-based low-resolution conformational sampling. In addition, it adds new functionality for the integration and translation between both levels of detail. The replica exchange method is implemented to allow enhanced sampling of both the all-atom and low-resolution models. The tool set aims at applications in structural biology that involve protein or nucleic acid structure prediction, refinement, and/or extended conformational sampling. With structure prediction applications in mind, the tool set also implements a facility that allows the control and application of modeling tasks on a large set of conformations in what we have termed ensemble computing. Ensemble computing encompasses loosely coupled, parallel computation on high-end parallel computers, clustered computational grids and desktop grid environments.

This paper describes the design and implementation of the MMTSB Tool Set and illustrates its utility with three typical examples—scoring of a set of predicted protein conformations in order to identify the most native-like structures, ab initio folding of peptides in implicit solvent with the replica exchange method, and the prediction of a missing fragment in a larger protein structure.

Keywords: Protein structure prediction; Replica exchange; Ensemble computing

1. Introduction

The success of computational methodologies in chemistry that have been developed over the last four decades is reflected in a multitude of academic and commercial programs available today. CHARMM [1], Amber [2], and Gaussian [3] are typical examples of this development, and enjoy wide usage in both academia and industry. Most of these programs that have emerged from this period are highly functional, well optimized, and sufficiently integrated within their intended range of applications. However, because of a high level of complexity, proprietary command interfaces and input/output formats these programs often tend to be inflexible when extensions and/or interoperability with other existing programs are needed. While this is a common problem in the integration of heterogeneous legacy software components [4], such issues have become especially apparent in the implementation of new enhanced sampling techniques applied to the conformational sampling of biopolymers. These novel simulation protocols combine existing methods in order to improve conformational sampling efficiency for molecular modeling and dynamics applications. Generalized ensemble sampling techniques, for example, involve parallel simulations of a system of interest with different weight factors coupled by a Monte Carlo simulation protocol [5–7]. Variants of this sampling scheme are being used increasingly in the study of long time scale phenomena such as protein folding [8–13]. These methods could be implemented in the form of separate, new programs or by modifying existing simulation packages. In a more practical implementation, however, existing programs could be used to run each of the simulations while an external interface layer is utilized to couple and control the individual simulations and facilitates the enhanced sampling methodology. This approach would allow greater flexibility in using the same enhanced sampling method with different simulation programs, and avoid difficulties in modifying existing large software packages directly.

Another way to improve conformational sampling is through multiscale modeling techniques. The computational modeling of biological macromolecules commonly revolves around structure representations in atomic or near-atomic detail, with a classical description of physical interactions.

[*] Corresponding author. Tel.: +1-858-784-8035; fax: +1-858-784-8688. *E-mail address:* brooks@scripps.edu (C.L. Brooks III).

1093-3263/$ – see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmgm.2003.12.005

Such models have been quite successful in complementing experimental data with structural, dynamic, and energetic information, but involve substantial computational resources for larger systems, or when long time scales have to be considered. In particular, studies of protein folding, structure prediction applications, or the formation and interaction of supramolecular assemblies become prohibitively expensive with models at atomic detail. Alternatively, coarser molecular representations with few virtual particles, often also projected onto lattices, have yielded meaningful results in such cases [14]. Unfortunately, the reduced level of detail often cannot provide the same accuracy as all-atom models. For example, it is quite feasible to generate native topologies from folding simulations using simple lattice models; however, it is much more difficult to actually discern native or near-native conformations from other, non-native conformations that are also generated with the same model [15]. In such cases, one may instead reconstruct all-atom structures from reduced representations [16], and use these more detailed models to regain a higher level of accuracy with an all-atom scoring function that can then distinguish native from non-native conformations [17,18]. This idea represents the core of more general multiscale modeling approaches; lower resolution models are used to extend sampling to longer time scales or larger system sizes, whereas higher-resolution models provide the energetic accuracy. While the structure prediction example above describes a single pass of low-resolution sampling followed by the use of all-atom models for improved accuracy, multiscale modeling can also be done in a continuous fashion, for example through Monte Carlo type simulations that repeatedly move between low- and high-resolution models for extended sampling on an energy landscape that is closely coupled to the interactions of the high-resolution model. The implementation of multiscale modeling methods faces problems similar to these seen in the implementation of enhanced sampling methods, but usually involves the combination of multiple programs rather than a single simulation program. All-atom modeling of biological macromolecules is possible with a number of standard molecular modeling packages such as CHARMM or Amber, but these programs usually do not fully support low-resolution models and especially lattice-based representations. On the other hand, simulation programs for low-resolution models, such as the lattice simulation program MONSSTER [19], do not usually allow all-atom modeling. Both types of applications are fairly complex, so that the option of simply merging them is not very attractive. As for the implementation of enhanced sampling methods, a better solution would be to wrap simulation programs for all-atom and low-resolution models through a common interface layer and provide translation routines between both models as the basis for building multiscale applications.

In this paper, we describe a new set of utilities and programming libraries for the implementation of computationally distributed enhanced and multiscale sampling methods based on existing simulation programs. This package, called Multiscale Modeling Tools for Structural Biology (MMTSB) Tool Set (available at http://mmtsb.scripps.edu/software/mmtsbToolSet.html), is an effort within the NIH Research Resource for Multiscale Modeling Tools for Structural Biology and follows the implementation strategy outlined above by integrating the existing programs through an interface layer while providing missing functionality as necessary. Interpreted scripting languages such as Perl or Python are particularly suitable for building interface layers since they combine ease of use and portability with a high level of functionality for addressing the complex system-oriented but computationally less intensive tasks [20]. Similar, scripting-language based designs have been used successfully in other related applications such as the molecular modeling tool kit (MMTK) [21] or the Bioperl toolkit [22].

The idea of the MMTSB Tool Set is not just to provide a set of user programs for certain enhanced and multiscale sampling modeling tasks, but also a programming workbench, which provides the framework for the development of new applications that require the interplay of multiple simulation packages. It focuses on applications in the area of protein structure prediction, protein folding, and large-scale model building and refinement of proteins and nucleic acids for which enhanced and multiscale sampling techniques are particularly useful. As a subset of its functionalities, the tool set also provides a common user interface to all-atom modeling via CHARMM[1] [1] or Amber[1] [2] and reduced-model lattice modeling via MONSSTER [19]. Furthermore, the tool set incorporates a number of support functions that are motivated by multiscale modeling applications, but are certainly useful for other purposes as well. They include algorithms for translating quickly and accurately between low- and high-resolution models and methods for the organization, manipulation, and evaluation of large sets of conformations for a given protein, in what may be referred to as ensemble computing. Ensemble computing applications greatly benefit from parallel execution since they are inherently parallel in nature and typically require relatively little communication. The tool set provides basic parallel platform support implemented on the scripting language level, which makes it largely platform-independent and does not require specific communication libraries.

In the following, we will first describe the architecture and components of the MMTSB Tool Set in more detail. We will then continue by providing examples of how the tool set may be used for typical enhanced and multiscale sampling applications in protein structure prediction, structure evaluation, and structure refinement examples. We conclude by discussing how this architecture may be extended to new tasks and applications.

[1] We note that the tool set is supported with versions of CHARMM beyond c29b and Amber version 7.

2. Software description

2.1. Architecture

Common modern scripting languages that would be appropriate for building complex applications are Perl and Python. We decided to use Perl as the (still) more widely used scripting language in order to minimize portability issues and to facilitate user extensions as much as possible. As depicted in Fig. 1, the architecture of the MMTSB Tool Set consists of a collection of object-oriented classes, called packages in Perl, that implement all of the core functionalities. These packages are used by a number of executable user programs, which mainly parse command line arguments and call the appropriate functions to build specific applications. In addition to the simulation programs CHARMM, Amber, and MONSSTER, a few other compiled-language programs are also included as part of the tool set, and wrapped through Perl, for computationally more demanding tasks that cannot be done efficiently in Perl alone. This program design maximizes flexibility and reusability. The packages alone can be used as a programming library for a variety of tasks that may go well beyond the intended applications of the MMTSB Tool Set. For example, one may use the interface to CHARMM to take advantage of Perl's advanced scripting capabilities for building complex modeling applications that require additional functionality and go beyond the capabilities of CHARMM's own scripting language. On the other hand, the command-line oriented user-level utilities in the MMTSB Tool Set are intended to cover a wide range of applications with a special focus on enhanced and multiscale sampling protocols. Furthermore, since these user-level utilities represent little more than a user interface to the package routines, they can easily be customized to address new types of problems either based on the existing package routines or by adding new functionality. The user utilities could also be replaced by a different type of user interface or integrated into other types of applications without significant additional effort.

Table 1
MMTSB Tool Set Perl packages implementing the core functionality and serving as a programming library

Package name	Functionality
Molecule	All-atom representation of molecule objects
CHARMM	Interface to CHARMM molecular modeling program
Amber	Interface to Amber molecular modeling program
Analyze	Structural analysis of molecular conformations
Cluster	Clustering of molecular conformations
SICHO	Reduced, side chain based representation of molecules
Sequence	Amino acid sequences and secondary structure information
MONSSTER	Interface to MONSSTER lattice simulation program
SimData, Ensemble	Ensemble computing
JobClient, JobServer	Parallel execution for ensemble computing applications
ReXClient, ReXServer	Replica exchange simulations
GenUtil	General utility functions
Server, Client	General server/client implementation

2.2. Components

In this section, we provide an overview over the various components of the MMTSB Tool Set. The packages and user programs are listed in Tables 1 and 2, respectively. This paper is meant to give an overview of the MMTSB Tool Set

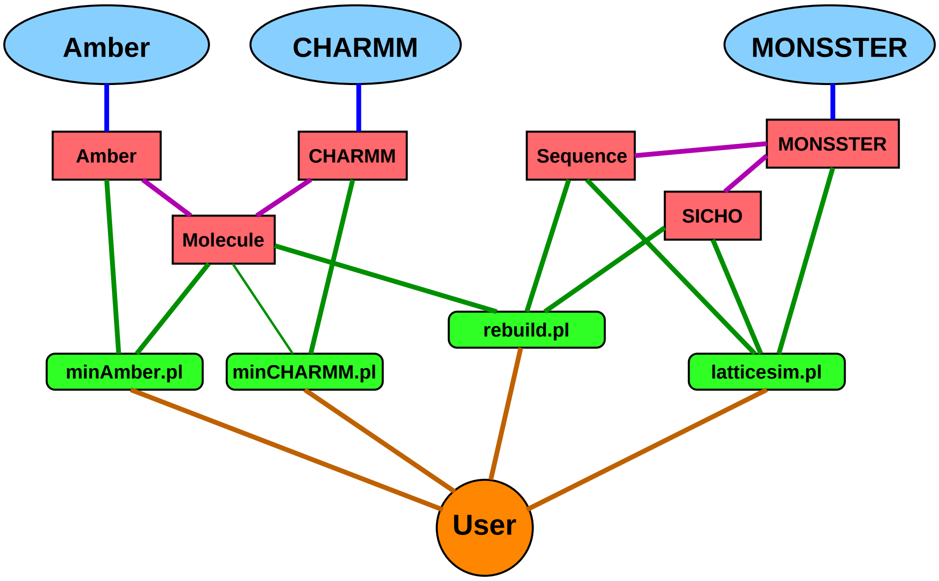

Fig. 1. Representative view of the MMTSB Tool Set architecture. External programs CHARMM, Amber, and MONSSTER are shown in blue, Perl packages in magenta, and Perl user utilities in green.

Table 2
Main MMTSB Tool Set command-line oriented user utilities

Utility	Functionality
enerCHARMM.pl, *enerAmber.pl*	All-atom energy evaluation with CHARMM/Amber
minCHARMM.pl, *minAmber.pl*	All-atom minimization with CHARMM/Amber
mdCHARMM.pl, *mdAmber.pl*	All-atom molecular dynamics with CHARMM/Amber
convpdb.pl	Convert and manipulate PDB files
complete.pl	Complete missing atoms in protein structures
mutate.pl	Mutate residues in protein structures
rms.pl, *lsqfit.pl*, *contact.pl*, *rgyr.pl*, *dihed.pl*, *qscore.pl*	Analysis of protein conformations
cluster.pl	Clustering of a set of conformations
latticesim.pl	Lattice-based low-resolution simulations with MONSSTER
genchain.pl	Generation of low-resolution representation from all-atom models
rebuild.pl	Reconstruction of all-atom models from low-resolution representations
ensmin.pl, *enseval.pl*, *enslatsim.pl*, *ensrun.pl*, *enscluster.pl*	Ensemble computing applications
checkin.pl	Create ensemble from external sources
getprop.pl, *setprop.pl*	Read/set ensemble property values
showcluster.pl, *bestcluster.pl*	Cluster-based analysis of ensemble data
aarex.pl, *aarexAmber.pl*	All-atom replica exchange simulations with CHARMM/Amber
latrex.pl	Lattice model replica exchange simulations with MONSSTER
rexinfo.pl	Extract replica exchange data
hlamc.pl, *hlamcrex.pl*	Hybrid multiscale lattice/all-atom Monte Carlo sampling protocol

predominantly from a user's perspective. The following description will focus on user programs rather than the underlying packages and support programs; however, some examples on how to use the packages as a library will be given as well. We will describe the basic functions for all-atom and low-resolution modeling and then continue with enhanced sampling and multiscale modeling applications.

3. All-atom modeling

The central part of the all-atom modeling components revolves around interfaces to the molecular mechanics packages CHARMM [1] and Amber [2]. In this respect the MMTSB Tool Set may be viewed as an alternative user interface to CHARMM and Amber for certain standard modeling tasks. The tool set utilities are meant to provide access to these powerful programs without requiring the user to go through the learning curve of understanding the specific command and data input and output protocols of each program. The functionalities that are provided through the MMTSB Tool Set focus on energy evaluations, minimization, and molecular dynamics runs with the utilities *enerCHARMM.pl*, *minCHARMM.pl*, *mdCHARMM.pl*, *enerAmber.pl*, *minAmber.pl*, and *mdAmber.pl*, respectively. Input structures are expected to be in standard PDB format with necessary name and format conversions done automatically and transparently for standard protein structures. A special utility, *convpdb.pl*, is also available for manual PDB format translations as well as a variety of manipulations that may involve changing residue numbering, editing chain identifiers, translating coordinates, and subselecting or merging structure fragments. All of the all-atom modeling utilities assume reasonable default values for a number of parameters, which can be altered by the user through additional command line options if necessary. For example, *minCHARMM.pl* without any further options will perform a short minimization in vacuum on a structure given as input and write the minimized structure to standard output. Parameters can then be altered to include implicit solvent [23,24], change the cutoff for non-bonded interactions, or the number of minimization steps, among other options. As another example, *mdCHARMM.pl* with default parameters will automatically recognize explicit water molecules in the input file in PDB format and setup and run a standard molecular dynamics protocol with periodic boundary conditions [25] and particle mesh Ewald electrostatics [26]. The same default usage will use implicit solvent based on a generalized Born formalism [27] instead, if explicit solvent molecules are not found. While many options are available to support a number of commonly used features in CHARMM and Amber, the MMTSB Tool Set does not aim to provide a complete interface to the full level of functionality of either one of these very complex molecular modeling programs. However, for modeling tasks that go beyond the capabilities of the provided utilities, the tool set may still be used to facilitate the preparation of input structures and setup procedures.

While the function of the MMTSB Tool Set as an interface to CHARMM and Amber may be very useful in itself, it should be emphasized again at this point that the real strength of the tool set lies in the combination of these basic all-atom modeling functions with other simulation techniques that are not available in CHARMM or Amber. These are in particular enhanced sampling facilities based on replica exchange methodology, multiscale modeling applications in a combination with low-resolution sampling, and ensemble computing techniques that allow the efficient application of a given modeling task to a large set of structures via distributed parallelism. Further aspects of this functionality will be described in more detail below.

4. Low-resolution modeling

Low-resolution modeling within the MMTSB Tool Set is based on the MONSSTER program [19]. MONSSTER implements the SICHO (side CHain only) model where each amino acid in a polypeptide chain is represented by a single virtual particle located at the side chain center of mass and projected onto a cubic lattice with 1.45 Å grid spacing [28]. Such a model is particularly well suited for constant temperature or simulated annealing type Monte Carlo simulations based on an energy function that is governed by physical and knowledge-based terms. As with CHARMM and Amber for all-atom modeling tasks, the tool set can also be used as a user interface for running low-resolution simulations with MONSSTER. The central utility for running either constant temperature or simulated annealing lattice simulations is *latticesim.pl*. Other supporting utilities are available for access to MONSSTER output files as well as the generation of sequence files, lattice chains, and other input files that are needed when running the MONSSTER program in a more manual fashion. Through the MMTSB Tool Set, lattice simulations can also benefit from enhanced sampling techniques and ensemble computing facilities. The latter is particularly useful for structure prediction applications where a very large number of structures are often generated with the fast lattice sampling protocol.

5. Translation between all-atom and low-resolution models

Both levels of detail, all-atom and low-resolution representations, are brought together by MMTSB functions that allow the generation of lattice chains from all-atom structures and the reconstruction of all-atom structures from lattice chains. Such mapping functions are essential for a multiscale modeling strategy and should preserve initial structures as much as possible through complete translation cycles. The utility for the generation of low-resolution models from all-atom structures is *genchain.pl*. It is primarily intended for generating lattice models suitable for MONSSTER, but it can also be used to generate related types of reduced models with or without additional particles at C_α positions, either in continuous space or projected onto cubic lattices with different grid spacings.

The reduction from all-atom models to low-resolution representations is fairly simple and straightforward. The reconstruction of all-atom models from low-resolution models, on the other hand, is more challenging since lost information has to be recreated by other means. Several methods are available for the reconstruction of complete all-atom models at moderate levels of accuracy based on C_α backbones [29–32]. In this case the backbone needs to be completed and side chains are typically added from a rotamer library and then annealed in order to resolve steric clashes. If the low-resolution model is side chain center based, one can use a slightly different reconstruction algorithm [16]. Because the side chain center is known, the reconstructed structures are generally quite close (<1 Å) to the original structure from which a low-resolution model was generated. The reconstruction program is part of the MMTSB Tool Set and available through the *rebuild.pl* utility. The rebuilding procedure can handle on- as well as off-lattice low-resolution models and is able to take advantage of C_α coordinates, if present, to build more accurate peptide backbones than would be possible with side chain centers alone.

As a first example how a combination of low-resolution and all-atom representations can be useful for common modeling tasks, one may consider the computational mutation of residues in a given protein structure. Off-lattice low-resolution models based on side chain centers and C_α coordinates can be used to preserve the backbone and the center of the original side chain while allowing the reconstruction of the mutated amino acid onto the same backbone at the same location. This may be done through a combination of the *genchain.pl* and *rebuild.pl* utilities, or more conveniently with *mutate.pl*, which is intended specifically for such computational mutation tasks.

6. Ensemble computing

Certain applications such as structure prediction, docking experiments, or estimates of conformational or interaction energies often involve relatively large ensembles of different conformations for a system of interest. Such ensembles may be assembled from simulation snapshots, the endpoints of simulated annealing runs as with the low-resolution lattice model described above, or by other means of conformational sampling. In many cases the ensemble structures are then evaluated and compared in one way or another, typically with the goal of extracting the most favorable ensemble members as the structures with the highest stability and, consequently, highest biological relevance. It may also be desirable to manipulate all of the ensemble structures in the same fashion in order to improve the evaluation process, for example by regularizing all of the conformations through force field based minimization.

The MMTSB Tool Set provides convenient facilities for handling structural ensembles in this manner. It allows the organization of ensemble members in the form of a simple database, along with associated properties such as energetic terms or structural quantities, and includes utilities for the application of the same operation on a whole ensemble of structures, in what we call ensemble computing.

Such computations are highly amenable to parallel computing environments, and the MMTSB Tool Set can take advantage of common architectures from distributed or shared memory parallel clusters to loosely coupled sets of heterogeneous machines through the use of a standard TCP/IP socket-based networking protocol [33]. With applications such as protein structure prediction in mind, special emphasis is put on tools that allow the efficient minimization and evaluation of energies based on CHARMM for an ensemble of structures. These functions are available with the utilities *ensmin.pl* and *enseval.pl*, respectively. A more general utility, *ensrun.pl*, allows any command or command script to be run on a set of structures in an ensemble either for calculating a property of interest or generating new sets of structures. The ensemble facility in the MMTSB Tool Set is designed to maintain multiple conformations for each member of an ensemble. Such sets of conformations may be derived through minimization, short molecular dynamics runs or other means of structure manipulation and they are identified through user-defined tags. This expands the ensemble idea borrowed from statistical mechanics to a collection of structures where each member is represented not just by one but by any number of related conformations. In this organizational scheme, multiple ensembles are only needed for entirely different sets of conformations or conformations that belong to a different system altogether. In these cases, multiple ensembles would be distinguished simply by keeping the data files in different subdirectories.

The internal organization of ensembles within the MMTSB Tool Set is illustrated in Fig. 2. A relatively simple, text file-based database setup was chosen to maintain a level of transparency and openness that allows easy access to stored conformations and properties with external programs. However, such a design comes at the expense of efficiency for very large structural ensembles. While significant limitations have not become obvious for applications involving up to 50,000 ensemble members, better performance for larger ensembles could be obtained through a more efficient database design. The use of database engines would be an option if performance improvements turn out to be necessary in the future.

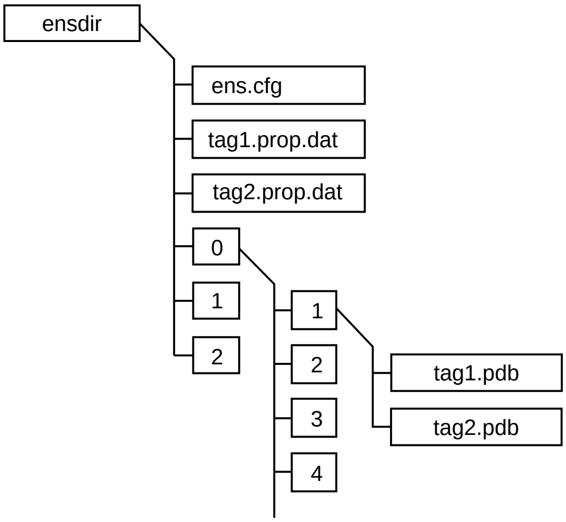

Fig. 2. Directory and file organization used for ensemble computing.

There are four ways for generating structure ensembles within the MMTSB Tool Set. The first option, aimed at structure prediction applications, generates ensembles from low-resolution lattice simulations with *enslatsim.pl*. This utility is an ensemble version of *latticesim.pl* taking advantage of parallel execution and allowing the automatic reconstruction of all-atom models from the final lattice models. In the second option one can generate ensembles from replica exchange simulations, which will be explained in more detail in the following section. For all other purposes, there is a general utility *checkin.pl* for creating new ensembles or adding one or more structures to an existing ensemble from external sources. Finally, as a fourth option, because of the simple database structure one may simply create an ensemble directory structure and copy files manually. This is not recommended, but it may be more practical in combination with other computational tools if integration within the MMTSB Tool Set is not possible or desirable.

Each set of structures in an ensemble has an associated property data file, which is queried most conveniently with the utility *getprop.pl*, but could also be easily read with other external programs, if necessary. The properties stored in this file are identified with arbitrary property tags and may be comprised of energy terms calculated with *enseval.pl*, structural properties calculated with *calcprop.pl*, or other properties resulting from external programs that are run with *ensrun.pl* over the whole ensemble. It is also possible to enter single values up to whole data series in a manual fashion with *setprop.pl*.

While some of the tools for ensemble computing are specifically aimed at multiscale modeling and structure prediction applications, the more general utilities make this kind of infrastructure accessible for other applications as well. It was our intention with this design that the MMTSB Tool Set will become useful for a variety of ensemble computing tasks that involve the organization and manipulation of large sets of molecular conformations in new contexts.

7. Replica exchange simulations

An exploration of the potential energy landscape for a system of interest, usually with the goal of finding low-lying regions, is the central theme of most molecular modeling applications. Sampling efficiency with standard simulation techniques such as molecular dynamics or Monte Carlo at a given temperature is governed by the distribution and height of energetic barriers, or ruggedness, and the slope towards the energy minimum in the landscape, both of which determine the kinetic behavior of the system. Barrier crossings are facilitated at higher temperatures, but a single simulation at an elevated temperature would sample an altered free energy surface due to temperature-dependent entropic contributions. As a dramatic example, a single simulation of a protein at a temperature above its folding temperature would eventually result in protein unfolding, since unfolded conformations have a lower free energy than conformations in the folded, native basin at such temperatures.

Enhanced sampling schemes have been introduced to address this problem, so that it becomes possible to overcome energetic barriers more easily while maintaining sampling on the relevant free energy surface at room temperature. In one such method, called replica exchange or parallel tempering, multiple simulations or replicas of the same system are run in parallel at different temperatures [12,34]. The individual simulations are then coupled through Monte Carlo based exchanges of simulation temperatures between replicas at periodic intervals. In this scheme each simulation visits a range from low to high temperatures so that sampling is provided at the temperature of interest, while traversing conformational space more easily at elevated temperatures.

More formally, temperatures are exchanged between two replicas, i and j, with temperatures T_i, T_j and energies E_i, E_j according to the canonical Metropolis criterion for the exchange probability p:

$$p = \begin{cases} 1 \quad \text{for } \Delta \le 0 \\ \exp(-\Delta) \quad \text{for } \Delta > 0 \end{cases}$$

where

$$\Delta = \left(\frac{1}{kT_i} - \frac{1}{kT_j}\right)(E_j - E_i)$$

Applied to one or more pairs of simulations after short runs of molecular dynamics or Monte Carlo simulations at constant temperature, this protocol can improve the sampling efficiency by orders of magnitude depending on the type and size of the system. Replica exchange simulations have been used with great success for the ab initio folding of peptides in explicit solvent from first principles [8–10,35]. In other applications, shorter replica exchange runs may be used for improved local structure refinement or simply for ranking a set of structures according to relative free energy, since the most favorable conformations will populate the lowest temperatures. While replica exchange simulations based on the exchange of temperatures have been most popular, other forms of biases can also be used to reweight sampling probabilities [6,7]. For example, umbrella type biasing potentials could be used to restrain the radius of gyration or the fraction of native contacts to different values in each replica and, in a more general case, multiple biases can be combined in two-dimensional or even higher-dimensional replica exchange simulations [36].

In the MMTSB Tool Set, replica exchange sampling is available to achieve enhanced sampling of all-atom models with CHARMM or Amber, as well as enhanced lattice based sampling of low-resolution representations with MONSSTER. In each case, the simulation control and exchange algorithm are implemented on the scripting language level. In fact, most of the same code is reused in these cases and could be combined easily with other applications in order to add replica exchange sampling. Replica exchange simulations are particularly suitable for parallel environments due to their inherent parallelism and low cost of communication, since communication occurs only infrequently at exchange events. As for the ensemble computing functions, the MMTSB Tool Set supports most parallel architectures and environments through its own platform independent communication protocol.

The main tools for running replica exchange simulations in the MMTSB Tool Set are *latrex.pl* for lattice-based replica exchange simulations using MONSSTER, and *aarex.pl* and *aarexAmber.pl* for all-atom replica exchange simulations using CHARMM and Amber, respectively. During and after a replica exchange run simulation data can be queried in many ways with the *rexinfo.pl* utility. Replica exchange simulations run through the MMTSB Tool Set involve a special directory structure for organizing and storing the conformations from each of the individual replicas, but an option is available to automatically build an ensemble data structure from the lowest temperature conformations for further processing with the ensemble computing tools.

8. Advanced multiscale sampling methods

The utilities for lattice-based low-resolution sampling, for all-atom sampling, and for the translation between low-resolution and all-atom models can be combined to implement a basic multiscale modeling protocol. This is provided with the utility *predict.pl*, which integrates these steps into a single pass from low-resolution sampling to all-atom based scoring for structure prediction applications. More complex multiscale modeling protocols, however,

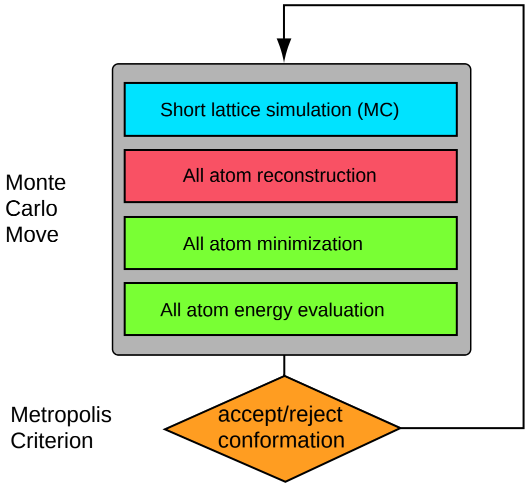
Short lattice simulation (MC)
All atom reconstruction
All atom minimization
All atom energy evaluation
Monte Carlo Move
Metropolis Criterion
accept/reject conformation

Fig. 3. Hybrid lattice/all-atom simulation scheme I.

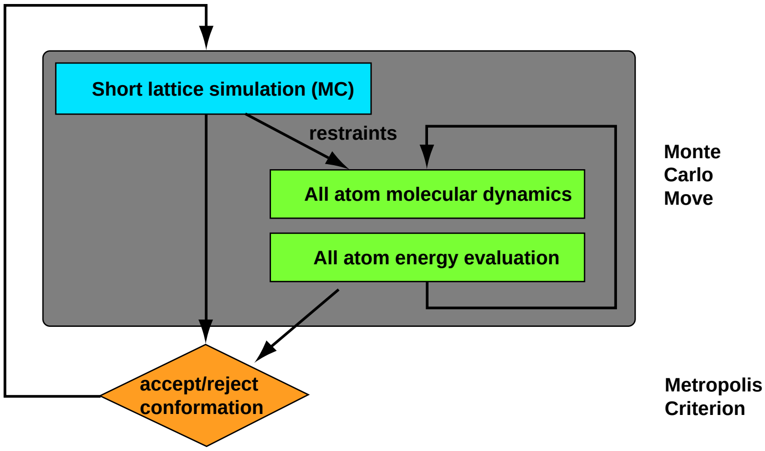
Short lattice simulation (MC)
restraints
All atom molecular dynamics
All atom energy evaluation
Monte Carlo Move
accept/reject conformation
Metropolis Criterion

Fig. 4. Hybrid lattice/all-atom simulation scheme II.

may involve the continuous transition between low and high-resolution models to take advantage of efficient sampling with the low-resolution model and an accurate energy function with all-atom models. While any protocol could be setup as a custom application using the MMTSB Tool Set packages or programming library, we have implemented two advanced modes of multiscale modeling simulations that appear to be particularly useful. Both of these sampling protocols follow the idea of sampling conformations on the all-atom energy landscape using the lattice model, and are implemented in *hlamc.pl*. In the first mode, illustrated in Fig. 3, a Monte Carlo simulation is run with moves that consist of a very short constant temperature lattice simulation followed by all-atom reconstruction and short all-atom minimization before the final all-atom energy is used in the Metropolis criteria. The second mode (Fig. 4) couples lattice and all-atom models more tightly by running short all-atom molecular dynamics simulations that follow conformational moves from lattice simulation through side chain center restraints. Again, the final energy is used in a Monte Carlo simulation to either accept and continue from favorable conformations or reject unfavorable conformations and try another move. Instead of a single Monte Carlo run, either mode can alternatively be coupled with replica exchange sampling at different temperatures with *hlamcrex.pl*. Other similar multiscale sampling algorithms are certainly possible, and the application utilities provided may serve as starting points for implementing new sampling schemes. While more testing and tuning of these novel methods is needed, we believe their availability through the MMTSB Tool Set will spark further interest.

9. Structure analysis functions

A number of utilities in the MMTSB Tool Set can be used for limited structure analysis tasks. They include functions such as clustering or the calculation of root mean square deviations and optimal superposition between two conformations, calculation of the radius of gyration, the fraction of native contacts, or standard peptide chain dihedral angles ϕ, ψ, ω, and χ_1. In ensemble computing applications, most of these structural properties can be calculated in parallel for a whole set of ensemble structures with the utility *calcprop.pl* in order to facilitate analysis of ensemble structures.

The clustering functions are particularly helpful for structure prediction tasks. Clustering is done based on pairwise distances, measured either as coordinate or dihedral angle root mean square deviations independent of any single reference structure. The results are sets of structures with similar conformations according to the given criteria. It is then possible to compare energy scores between entire clusters as the average score from all of their respective members and obtain statistically more reliable quantities such as energy scores from a cluster of similar conformations rather than single conformations. This type of analysis is facilitated for ensembles with the *enscluster.pl* and *bestcluster.pl* utilities for the generation of clusters and cluster-based analysis, respectively.

10. Applications

Having provided an overview of the different components of the MMTSB Tool Set, we now present a few typical applications that illustrate the use of the tool set—scoring of previously generated protein conformations with the ensemble computing facility, folding of peptides via replica exchange simulations, and the prediction of a missing fragment in the context of a known structure.

10.1. Scoring of protein conformations

The energy based scoring of protein conformations is a common task in structure prediction and docking protocols. In these cases the scoring function is typically applied to a large number of conformations generated with a given sampling method, with the goal of finding the most favorable, and presumably most native-like, structures. The ensemble computing facilities within the MMTSB Tool Set are particularly well suited for such a task. The following example illustrates the use of the MMTSB Tool Set for scoring predictions for the structure of a fusagenic sperm protein from H. fulgens [37]. Predictions of this protein were submitted during CASP4 (target id: T0125), the fourth community-wide assessment of structure prediction methods. For the example shown here, we downloaded all of the predictions for the entire length of the sequence submitted as the first model by participating prediction groups from the CASP web site. This yielded a total of 90 structure predictions.

11. Generation of an ensemble data structure from input files

As the first step, an ensemble is generated from the set of predicted structures by using the *checkin.pl* utility:

```
checkin.pl casp T0125*.pdb
```

The file names of the predictions submitted to CASP follow the format T0125*.pdb for this target and the structures are given an identifying tag *casp* in the newly created ensemble. Now that the predicted structures are available in ensemble format, ensemble computing tools can be used for further processing.

12. Preprocessing of input structures

Depending on how the input structures were generated, it is often a good idea to regularize and minimize the structures before calculating energy scores. Many structure predictions do not contain a complete set of atoms. Often, hydrogen atoms are missing and some predictions may consist only of C_α coordinates. Therefore, as the first step we will run the *complete.pl* utility in order to generate complete, all-atom structures for all of the predictions. Since we want to apply this command to all of the structures in the ensemble we use *ensrun.pl* as follows:

```
ensrun.pl -new caspcomplete casp complete.pl
```

This command runs the *complete.pl* utility for each structure in the ensemble under the *casp* tag, the only set of structures we have so far, and generates a new set from the output of *complete.pl* with the tag *caspcomplete*. The *complete.pl* utility automatically uses different protocols depending on how much of the structural information is missing. If only C_α or backbone coordinates are present, the SCWRL utility [30,38] is used to add side chains from a rotamer library, while missing hydrogens are added with the HBUILD facility in CHARMM [1].

The next step is a short minimization run with a distance dependent dielectric function. This can be done very conveniently with the *ensmin.pl* utility which uses CHARMM to do the actual minimization:

```
ensmin.pl -cpus 4 -par minsteps=100,dielec=rdie,epsilon=4.0 caspcomplete min
```

This command minimizes all of the completed structures stored under the *caspcomplete* tag and creates a new set of structures with the *min* tag. In this example 100 steps of minimization are requested with a distance dependent dielectric function and $\varepsilon = 4$. Depending on the size of the system, minimization runs can take some time to complete and it is advantageous to use parallel computing facilities to speed up the calculation. In this example four CPUs are used in a shared memory environment.

13. Evaluation of scoring function

Finally, we can evaluate a scoring function for the minimized structures. The ensemble computing tool for energy evaluation, *enseval.pl*, is used as follows:

```
enseval.pl -cpus 4 -set score=total -par gb min
```

Here, we are using a scoring function that includes implicit solvation based on a generalized Born formalism [27], in this case the GBMV method [39,40], as implemented in CHARMM as the default when GB is requested. Again, four CPUs are used in parallel to speed up the calculation. In this case the total energy of the entire molecular mechanics force field, including all bonded and non-bonded interactions as well as the electrostatic solvation term, are used as the scoring function and assigned to a new property called *score*.

14. Analysis of results

Once the *enseval.pl* run is complete the scores are available and can be queried with *getprop.pl*. A sorted list of all values is obtained easily with the following command:

```
getprop.pl -prop score min | sort +1n


78  -6948.125290
77  -6948.045810
80  -6923.875040
81  -6922.909130
...
```

In this command the property name *score* and structure tag *min* are used to identify the data set. Such a result may be sufficient for many applications, but often it is advantageous to form clusters of input structures based on mutual similarity and then compare average scores over cluster members to identify the lowest scoring clusters rather than individual structures. This requires a few additional steps and will be illustrated in more detail in the loop prediction example.

Since the native conformation of this structure is available from the protein data bank [41] (PDB code: 1GAK) it is possible to calculate root mean square deviations (RMSD) between all of the predictions and the native structure. This can be done conveniently with *calcprop.pl*, which calculates a number of structural properties, including RMSD:

```
calcprop.pl -natpdb 1gak.pdb min
```

Both, the energy score and C_α RMSD values, can now be extracted with

```
getprop.pl -prop rmsdca,score min

1 4.236713 -6607.168500
2 5.965288 -6206.241270
3 4.499256 -6398.912490
4 7.499939 -6505.901710
...
```

The results are visualized in Fig. 5 as a plot of energy scores versus root mean square deviations from the native structure. It can be seen that the energy scores decrease on average towards more native-like conformations, and the scoring function could be indeed used to identify the most native like conformations, with a C_α RMSD of about 4 Å in this example. In a recent study, we have applied similar protocols to all of the predictions submitted to CASP4 and generally found good correlation between all-atom energy scores that include a realistic treatment of solvation and proximity of predicted conformations to the experimentally obtained native structure [17].

14.1. Folding of peptides with replica exchange simulations

The MMTSB Tool Set adds enhanced sampling capabilities to existing simulation programs such as CHARMM or Amber through the replica exchange simulation methodology. Replica exchange simulations can speed up sampling in conventional molecular dynamics simulations by orders of magnitude [8,12]. Such a gain in sampling efficiency is particularly attractive for the challenging problem of folding peptides and proteins through simulation at atomic detail. Ab initio folding at atomic detail has been simulated directly with constant temperature molecular dynamics simulations only for very small peptides, where folding times are on the order of hundreds of nanoseconds and considerable computational resources were used [42,43]. When replica exchange simulations are employed, ab initio folding of peptides can be achieved for larger systems and on much shorter timescales [9,10,35]. Further reduction of computational expense is possible if an implicit solvent description is used instead of explicit solvent molecules [44–47]. It then becomes possible to fold α-helices and β-hairpins in a matter of days with moderate computational resources. As examples we will consider the peptide $(AAQAA)_3$, which is known experimentally to be predominantly α-helical [48], and the designed tryptophan zipper hairpin SWTWENGKWTWK [49], for which an experimental structure is available from NMR. A replica exchange simulation with the MMTSB Tool Set starting from a completely extended conformation for either peptide is run from the command line as follows:

```
aarex.pl -temp 8:270:550 -mdpar gb,cmap,scalerad,gbmvsa=0.012,dynsteps=500,blocked
-n 10000 extended.{aaqaa/hairpin}.pdb
```

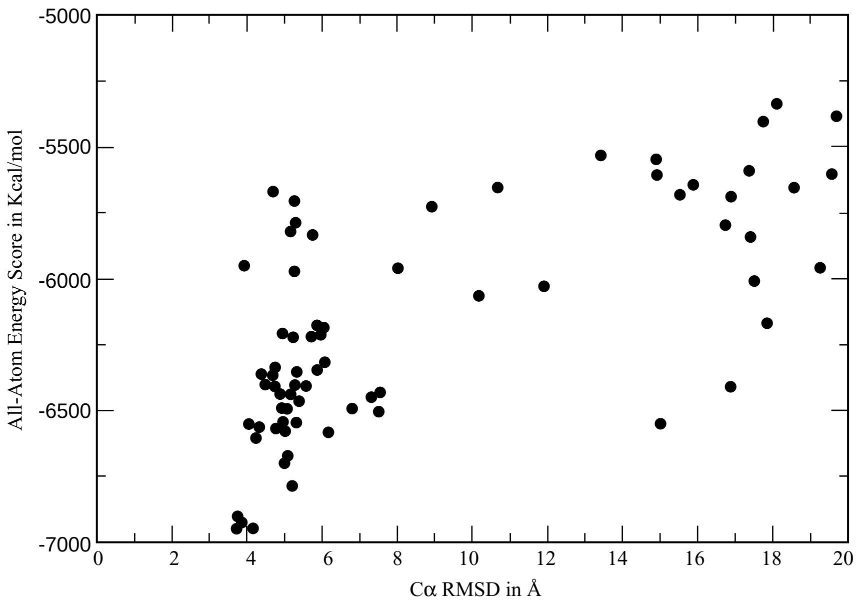

Fig. 5. All-atom energy scoring function versus root mean square deviation (RMSD) of C_α coordinates with respect to native structure (PDB code: 1GAK). The all-atom energy function based on the CHARMM22 force field [56] includes implicit solvent contributions from the GBMV implementation of the generalized Born formalism [39,40]. Data points with energies more positive than −5000 kcal/mol and more than 20 Å RMSD are omitted for clarity.

In these simulations we are using the GBMV implementation [39,40] of the generalized Born formalism with modified van der Waals radii based on the set of radii developed by Nina et al. [50] and a hydrophobic term that depends linearly on the solvent accessible surface area with a scaling factor of 0.012 kcal/mol/Å^2. We also use recently developed map-based ϕ/ψ backbone dihedral cross terms, in order to adjust the balance between α-helical and extended conformations from the original force field [51]. Both peptides are built with blocked termini. The eight temperature windows are exponentially spaced from 270 to 550 K resulting in temperature exchange probabilities between replica pairs from 16 to 19% for the hairpin and from 16 to 25% for the helical system. The simulations are each carried out over a total of 10 ns simulation time for each replica (10,000 cycles and 500 molecular dynamics steps (step size: 2 fs) between exchanges), which takes about 1 week on eight nodes of a PC-based cluster.

The data from replica exchange simulations can be examined with the *rexinfo.pl* utility and conformations may be analyzed with *rms.pl* and *genseq.pl*. Particularly useful is an option to rank replicas according to the average temperatures they have visited over a period of simulation time. This information can be used to identify the most favorable conformations from the replica with the lowest average temperatures. Tables 3 and 4 summarize the results from simulations for $(AAQAA)_3$ and the hairpin system, respectively. The data shows that the most favorable final conformations at the lowest temperatures agree well with experimental observations. Replica 7 in the simulation of $(AAQAA)_3$, with the lowest average temperature at the end of the simulation, is indeed α-helical (see also Fig. 6). On the other hand, the most favorable replica in the hairpin simulation, replica 1, clearly exhibits a hairpin-like secondary structure and deviates by only 1.4 Å RMSD from the experimentally determined structure. However, in this example we also find other structures at higher temperatures that contain partial α-helices. The excellent agreement of the simulated hairpin from replica 1 is also manifest in Fig. 6, where the final conformation is compared with the structural ensemble from NMR measurements [49].

It is instructive to examine the evolution of RMSD and temperature for replica 7 in the $(AAQAA)_3$ simulation (Fig. 7a) and replica 1 in the hairpin simulation (Fig. 7b).

Table 3
Results from replica exchange simulations of $(AAQAA)_3$

Replica	Temperature rank	Time spent at 270 K	Secondary structure AAQAAAAQAAAAQAA
7	1.4	67.8%	.HHHHHHHHHHHH..
8	2.5	24.8%	....HHHHHHHHHH.
6	3.0	4.8%	........HHHHHH.
4	3.3	2.6%	.HHHHHHHHH.....
5	6.1	0.0	....
2	6.4	0.0	....
3	6.6	0.0	....
1	6.8	0.0	....

Temperature rank and percentage of time spent at the lowest temperature, 270 K, are averaged over the last 1000 cycles (9000–10,000). The secondary structure is obtained from the final conformation after 10,000 cycles using the DSSP program [55].

Table 4
Results from replica exchange simulations of the harpin sequence SWTWENGKWTWK

Replica	Temperature rank	Time spent at 270 K	Secondary structure SWTWENGKWTWK	Cα RMSD in Å
1	1.1	87.8%	.EEEE..EEEE.	1.4
6	2.2	10.9%	....	4.7
8	3.8	0.9%	.HHHHH......	5.7
7	4.1	0.4%	....	3.9
5	5.3	0.0	....	6.9
2	5.7	0.0	....	3.3
4	6.7	0.0	....	5.9
3	7.2	0.0	....	5.5

Temperature rank and percentage of time spent at the lowest temperature, 270 K, are averaged over the last 1000 cycles (9000–10,000). The secondary structure is obtained from the final conformation after 10,000 cycles using the DSSP program [55]. C_α coordinate root mean square deviations are calculated with respect to the native structure obtained from NMR experiments (PDB code 1LE1, model 1) [49].

In both cases, native-like structures are reached relatively quickly (after 5 and 2.5 ns, respectively). The variations in temperature suggest extensive conformational sampling at higher temperatures during the beginning of the simulation until a favorable, native-like conformation is found and then the temperature remains at the lower temperatures. For the hairpin, the step-wise reduction of RMSD with respect to the native structure can be correlated with the temperature fluctuations. Major transitions at 1.8 and 2.5 ns appear to occur during or shortly after brief periods of elevated temperatures around 500 K when the crossing of conformational barriers is greatly facilitated.

This example is meant to demonstrate how the MMTSB Tool Set can be used to take advantage of the replica exchange enhanced sampling methods in combination with modeling packages such as CHARMM and Amber. We hope that it will enable further studies in peptide folding, protein folding, and protein structure refinement.

14.2. *Prediction of missing fragments in proteins*

The large number of solved experimental protein structures provides the basis for finding at least partial templates in most structure prediction applications based on sequence homology or fold recognition. This reduces typical structure prediction efforts from entirely *de novo* predictions to the still challenging task of modeling unknown structural fragments in the context of a template. In principle, a multiscale

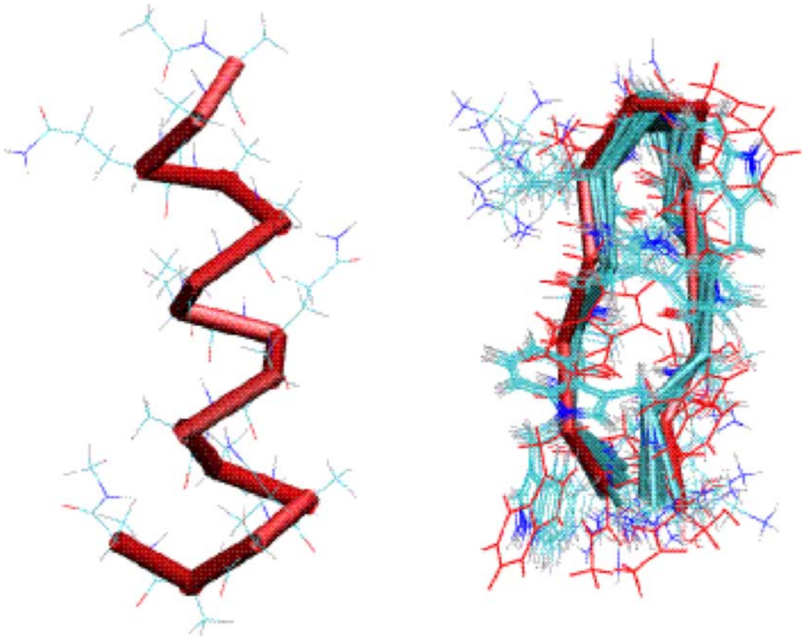

Fig. 6. Final conformations from lowest temperature replicas after 10,000 cycles (10 ns) in simulations of $(AAQAA)_3$ (left) and the hairpin sequence SWTWENGKWTWK (right). The hairpin structure is compared with the first 10 NMR models from PDB entry 1LE1 [49].

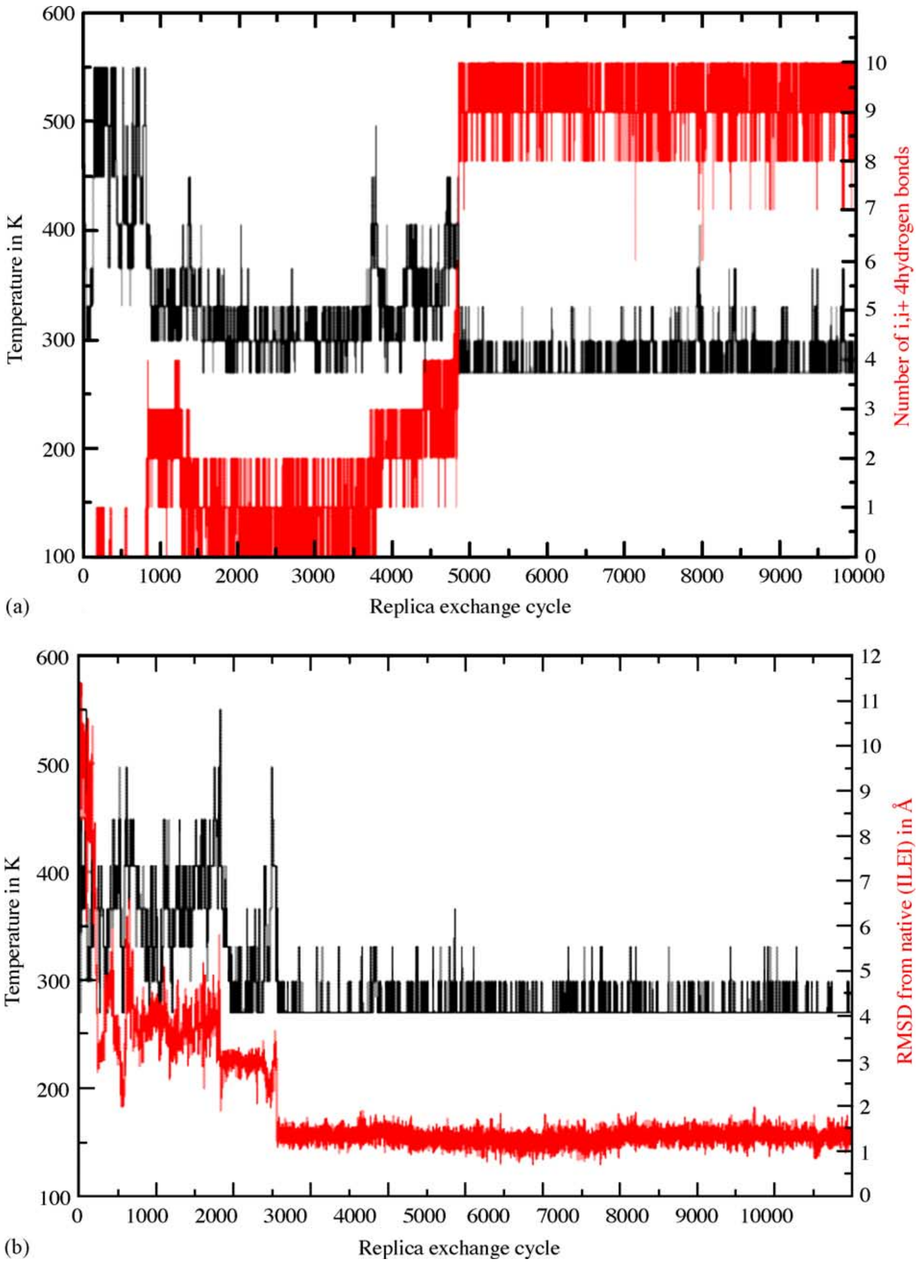

Fig. 7. (a) Time series of simulation temperature and number of α-helical $i, i+4$ hydrogen bonds in replica 7 over the course of the simulation of $(AAQAA)_3$. (b) Time series of simulation temperature and C_α RMSD with respect to the native structure (PDB: 1LE1) in replica 1 over the course of the simulation of SWTWENGKWTWK.

modeling protocol as outlined above is followed, but with the restraint of keeping the template structure fixed and the possibility to limit calculations to the vicinity of a modeling region and reduce computational expenses. How this can be done efficiently with the MMTSB Tool Set will be illustrated in the following example, which demonstrates the modeling of residues 48 to 55 in the zinc endopeptidase astacin from European fresh water crayfish (PDB code: 1IAB). The native structure is a mixed α/β-fold, and the missing 8-residue piece constitutes a long, solvent-exposed loop between two β-sheet segments.

15. Generation of model conformations from lattice simulations

In the first step, conformations for the missing fragment are generated using lattice-based low-resolution sampling. As input for this step only a sequence file and the template structure are needed. The sequence file contains the entire sequence for the template as well as the missing part. It also provides secondary structure information that is trivially obtained for the template and can be predicted for the missing part with good reliability using a variety of different secondary structure prediction methods.

In this example, we will use replica exchange simulations with the lattice model for enhanced sampling. The command would then look like this:

```
latrex.pl -n 1000 -temp 8:0.9:1.6 -l 48:55 -ensdir ens -ens lat -par seq=1iab.seq
1iab.incomplete.pdb
```

This command will run 1000 cycles of replica exchange on eight CPUs, which takes on the order of 1 h on modern clusters. With this command, all-atom structures are automatically rebuilt from the lowest-temperature replicas and stored as an ensemble for further processing. Since the −l option is given with the residues of the unknown structural fragment, only these residues are sampled freely, while the rest of the structure is restrained harmonically to the positions from the incomplete input structure that is used as a template. The structures sampled at the lowest temperature at each cycle are used to build an ensemble data structure in the directory *ens*.

16. Selection of protein environment near region of interest

The sampled structures could now be minimized, scored, and analyzed as in the example above. With 200 residues, the complete protein is fairly large, and both the minimization and energy evaluation steps are relatively expensive. Since only a small part of the structure has been varied in the sampling protocol, one does not necessarily need to consider the entire structure. The part of the structure in the vicinity of the variable residues can be cut out according to a distance cutoff similar to the range of electrostatic interactions, for example 12 Å. While this would very likely result in broken peptide chains at the edge of the cutout region the structure can be kept intact if relatively strong restraints are applied to fix the outer layer residues in place. This is in the spirit of the stochastic boundary approach to simulating localized regions of biopolymer structures [52]. Strains between highly restrained residues and entirely unrestrained parts may be relieved further if a second, intermediate layer of weakly restrained residues is introduced. This setup then results in a system where the variable residues that are being modeled are entirely flexible, surrounded by a first layer of weakly restrained residues and a second layer of highly restrained residues. Depending on the size of the system and the chosen cutoff this may result in significant computational advantages if only part of a large system needs to be considered.

The MMTSB Tool Set offers the utility *enscut.pl* to cut out such regions for an ensemble of structures and automatically setup the necessary residue restraint lists. It creates the list of residues that are included in the cutout region based on all of the different conformations for the variable residues as found in the ensemble, so that the same residues are cutout for each ensemble structure and energy values calculated at a later point remain comparable.

In our example, we will use a cutoff of 12 Å for including residues at all and a cutoff of 9 Å for residues that are weakly restrained:

```
enscut.pl -l 48:55 -hard 12 -soft 9 lat latcut
```

The cutout structures are then available under the tag *latcut* in the same ensemble and can be used for further processing. In this example the original structure with 200 residues is reduced to a region of interest of 123 residues, which translates into significant time savings in subsequent steps.

17. Scoring of conformations

Following the example above, the sampled conformations are first minimized before being scored with an energy function that includes implicit solvation based on the generalized Born formalism [39,40]:

```
ensmin.pl -cpus 4 -par minsteps=100,dielec=rdie,epsilon=4.0 -opt ens/latcut.options
latcut cutmin

enseval.pl -cpus 4 -set score=total -par gb cutmin
```

In this case, we create a minimized structure under the tag *cutmin*. Options specifying restraints to keep the cutout region intact during the minimization as described above are read from an options file generated automatically by *enscut.pl* when the structures were reduced.

18. Clustering and analysis

At this point energy scores are available for the sampled conformations and we can proceed to cluster the sampled conformations based on mutual root mean square deviations with the command *enscluster.pl*:

```
enscluster.pl -l 48:55 cutmin
```

Since we are primarily interested in the conformation of residues 48 to 55, we will cluster only based on these residues, disregarding the surrounding template. A quick view of the resulting clusters is available with the command *showcluster.pl*:

```
showcluster.pl cutmin


t              1000     7
  t.1            84     0
  t.2            44     0
  t.3           155     0
  t.4           435     0
  t.5           187     0
  t.6            38     0
  t.7            57     0
```

In this example, we find 7 clusters with sizes ranging from 38 to 435 members. Fig. 8 shows the sampling for this example with the experimental native structure as the reference for the RMSD calculations.

The utility *getprop.pl* could now be used with each cluster to obtain average energy scores and rank clusters accordingly. However, this can be done more conveniently with the *bestcluster.pl* utility, which automatically calculates average scores, standard deviations, and statistical errors for all clusters and ranks them accordingly. It can also calculate averages for only the subset of structures where the scores fall within two standard deviations of the mean. This is helpful when outliers occur with very high energies due to steric clashes that could not be resolved in the minimization procedure, and is used in the following example:

```
bestcluster.pl -prop score -crit avglow cutmin


t.3            155   122   -1120.1448  183.6851   16.6301
t.5            187   133   -1021.0567  219.7689   19.0564
t.2             44    39   -1006.8932  218.6557   35.0129
t.1             84    65    -970.0093  225.2203   27.9351
t.6             38    32    -827.8118  207.4521   36.6727
t.4            435   327    -760.3633  290.0010   16.0371
t.7             57    51    -508.4473  843.4406  118.1053
```

We find that cluster t.3 (colored red in Fig. 8) has the lowest average score (column 4) and the statistical error of 16.6 kcal/mol (column 6) indicates that the difference of approximately 100 kcal/mol to the next best cluster t.5 (colored blue in Fig. 8) is significant. The second column shows the total number of conformations in a given cluster. Column three indicates how many were actually used to compute the average, while the remaining conformations with much higher energy scores were excluded. The results confirm the qualitative picture in Fig. 9 of a downward slope of average energy towards more native-like structures. It should be stressed, though, that in a real structure prediction application the RMSD values with respect to the native conformation would obviously not be available.

The lowest energy conformation of the best cluster t.3 can then be found by calling *getprop.pl* with the cluster name as an additional argument. Finally, one may want to merge the final conformation with the original template in order to regain a complete protein structure. This can be done with the *convpdb.pl* utility and may be followed by a quick minimization run with restrained C_α atoms to anneal the merged structure.

In this case, the cluster with the conformations generated from the lattice protocol that are closest to the native structure was easily identified with this multiscale sampling protocol. While the best conformations in this cluster are close to 2 Å RMSD from the native, the conformation with the lowest energy score is found with an RMSD value of 3.6 Å. This structure, shown in Fig. 9, has the correct loop conformation for the most part but is shifted somewhat with respect to the experimental structure. At this point further sampling and refinement could focus only on structures from the best cluster in order to better distinguish structures closest to the native conformation.

18.1. Programming interface

The user-level utilities provide a comprehensive set of functions for enhanced and multiscale sampling applications; however the MMTSB Tool Set can also be used as a programming library for new tasks that involve or combine all-atom modeling, low-resolution modeling, and enhanced sampling methods. Such new applications would have to be written in Perl in order to take full advantage of the Perl packages from the tool set, but it is always possible to include components from other compiled or scripting languages through wrappers. One may also wrap the Perl scripts if they are to be used in other scripting environments although this may not always be an efficient solution.

As an example demonstrating the use of the tool set packages as a programming library for new Perl scripts, let us consider the analysis of secondary structure for low-resolution lattice chains based on backbone dihedral angles in rebuilt all-atom structures. This is done with the following steps: First, the SICHO chain file in MONSSTER format containing the input lattice structure is read. Since the lattice chain does not contain any sequence information, we need to read a sequence file, also in MONSSTER format, for this example. An all-atom molecule object is then rebuilt from the SICHO lattice chain. The ϕ and ψ dihedral angles can then be calculated, analyzed, and written out to standard output. The corresponding script could be as follows:

```
#!/usr/bin/env perl

use Molecule;
use Sequence;
use SICHO;
use Analyze;

my $seqfile=shift @ARGV;
my $chainfile=shift @ARGV;

my $chain=&SICHO::new();
$chain->readMONSSTER($chainfile);

my $seq=&Sequence::new();
$seq->readMONSSTER($seqfile);

my $mol=&Molecule::new();
$mol->rebuildFromSICHO($seq,$sicho);

my ($phi,$psi)=&Analyze::dihedral($mol);

my $molres=$mol->{chain}->[0]->{res};
for (my $i=1; $i<$#{$molres}; $i++) {
  my $sectype=”other”;
  $sectype=”beta” if ($phi->[$i]<0 && $psi->[$i]>90);
  $sectype=”alphaR” if ($phi->[$i]<0 && $psi->[$i]<60 && $psi->[$i]>-40);
  $sectype=”alphaL” if ($phi->[$i]>0 && $psi->[$i]>-40 && $psi->[$i]<50);

  printf “%3s %3d %5.0f %5.0f %s\n”,
  $res->[$i]->{name},$res->[$i]->{num},$phi->[$i],$psi->[$i],$sectype;
}
```

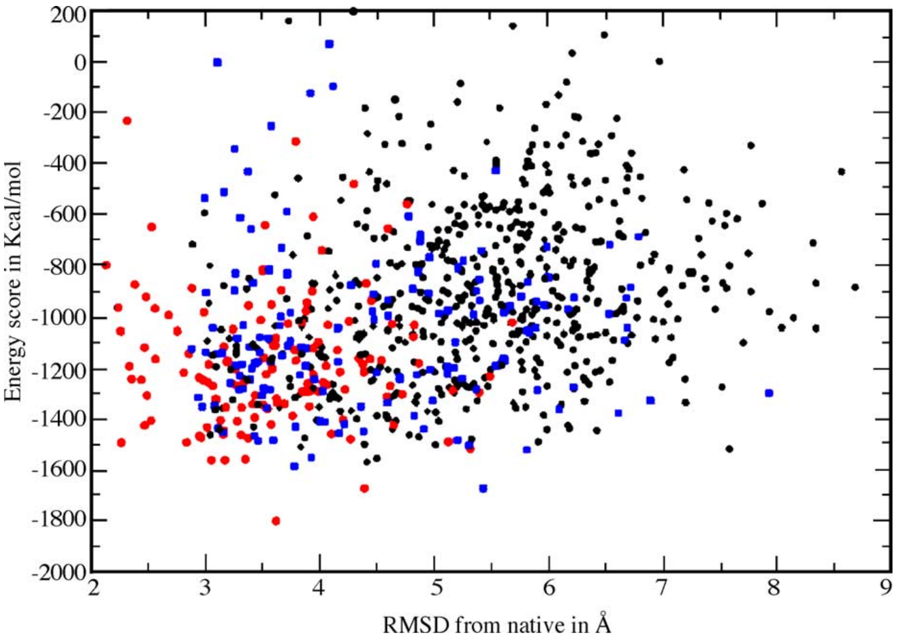

Fig. 8. Energy score including implicit solvent vs. RMSD from native conformation for loop residues 48–55 in astacin (PDB code: 1IAB) for structures generated with lattice sampling protocol. The cluster with the lowest average energy score, t.3, is colored in red, the second best cluster, t.5, is colored in blue.

This script uses four packages from the MMTSB Tool Set and would require significantly more effort if it had to be written without using the functionality of the tool set. When this example is run, it expects a sequence and chain file as input and writes out a list of residues with their corresponding ϕ/ψ angles and assigned secondary structure types.

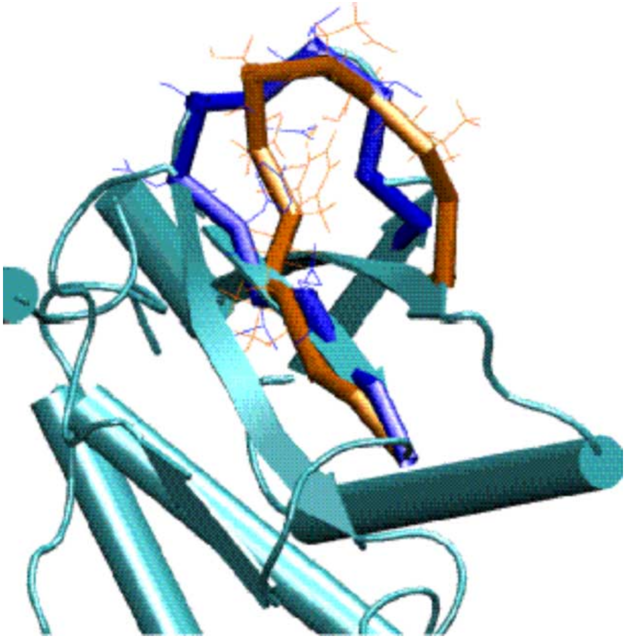

Fig. 9. Predicted loop conformation (orange) with lowest score from best cluster (t.3) compared with experimental structure (blue).

19. Summary

We have introduced the MMTSB Tool Set, a collection of utilities and programming libraries aimed at enhanced sampling and multiscale modeling applications in structural biology. The tool set interfaces with the standard molecular modeling packages CHARMM and Amber for all-atom modeling and with MONSSTER for low-resolution lattice-based simulations. It adds a number of functions, such as the translation between all atom and low resolution representations, and implements replica exchange sampling both for all-atom and lattice-based simulations. Another feature the MMTSB Tool Set enables ensemble computing for the application of programs and functions to large sets of structures. The MMTSB Tool Set is intended primarily to address problems in protein structure prediction, but it also serves as a simplified interface to the complex modeling packages CHARMM, Amber, and MONSSTER and we certainly hope that it will become useful for other applications as well.

We have presented three illustrative examples of how the MMTSB Tool Set may be used. While the examples present real cases, they are not intended to validate the methods that were being used. While a more careful evaluation of the methodology has been ongoing [16,17,51,53,54] and will be continued in the future, the purpose of this paper is to demonstrate the capabilities of the tool set.

Future developments of the MMTSB Tool Set may expand the availability of new enhanced sampling methods, implement more advanced multiscale sampling algorithms, and offer an alternative graphics-based user interface.

Acknowledgements

We thank the first users of the MMTSB Tool Set for testing and useful suggestions. Financial support from the NIH supported resource Multiscale Modeling Tools for Structural Biology (http://mmtsb.scripps.edu) (grant RR12255 to CLB III) is acknowledged. Support for the computational infrastructure and personnel (MF) that enabled these calculations came from the DOD through the grant DAMD17-03-2-0012 and is greatly appreciated.

References

[1] B.R. Brooks, et al., CHARMM: a program for macromolecular energy, minimization, and dynamics calculations, J. Comput. Chem. 4 (1983) 187–217.

[2] D.A. Pearlman, et al., AMBER, a package of computer programs for applying molecular mechanics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules, Comput. Phys. Commun. 91 (1995) 1–41.

[3] M.J. Frisch, et al., Gaussian 98. Gaussian, Inc, Pittsburgh, PA, 1998.

[4] A.C. Siepel, et al., An integration platform for heterogeneous bioinformatics software components, IBM Syst. J. 40 (2001) 570–591.

[5] U.H.E. Hansmann, Generalized ensemble techniques and protein folding simulations, Comput. Phys. Commun. 147 (2002) 604–607.

[6] U.H.E. Hansmann, New algorithms and the physics of proteins, Phys. A 321 (2003) 152–163.

[7] T. Nagasima, et al., Generalized ensemble simulations of spin systems and protein systems, Comput. Phys. Commun. 146 (2002) 69–76.

[8] K.Y. Sanbonmatsu, A.E. Garcia, Structure of met-enkephalin in explicit aqueous solution using replica exchange molecular dynamics, Proteins 46 (2002) 225–234.

[9] R. Zhou, B.J. Berne, R. Germain, The free energy landscape for β hairpin folding in explicit water, Proc. Nat. Acad. Sci. U.S.A. 98 (2001) 14931–14936.

[10] A.E. Garcia, K.Y. Sanbonmatsu, Exploring the energy landscape of a β hairpin in explicit solvent, Proteins 42 (2001) 345–354.

[11] Y.M. Rhee, V.S. Pande, Multiplexed-replica exchange molecular dynamics method for protein folding simulation, Biophys. J. 84 (2003) 775–786.

[12] Y. Sugita, Y. Okamoto, Replica-exchange molecular dynamics method for protein folding, Chem. Phys. Lett. 314 (1999) 141–151.

[13] D. Gront, A. Kolinski, J. Skolnick, A new combination of replica exchange Monte Carlo and histogram analysis for protein folding and thermodynamics, J. Chem. Phys. 115 (2001) 1569–1574.

[14] J. Skolnick, A. Koliniski, A unified approach to the prediction of protein structure and function, Adv. Chem. Phys. 120 (2002) 131–192.

[15] H. Lu, J. Skolnick, A distance-dependent atomic knowledge-based potential for improved protein structure selection, Proteins 44 (2001) 223–232.

[16] M. Feig, et al., Accurate reconstruction of all-atom protein representations from side-chain-based low-resolution models, Proteins 41 (2000) 86–97.

[17] M. Feig, C.L. Brooks, III, Evaluating CASP4 predictions with physical energy functions, Proteins 49 (2002) 232–245.

[18] C. Simmerling, et al., Combining MONSSTER and LES/PME to predict protein structure from amino acid sequence: application to the small protein CMTI-1, J. Am. Chem. Soc. 122 (2000) 8402–8932.

[19] J. Skolnick, A. Kolinski, A.R. Ortiz, MONSSTER: a method for folding globular proteins with a small number of distance restraints, J. Mol. Biol. 265 (1997) 217–241.

[20] D.M. Beazley, An extensible compiler for creating scritable scientific software, in: Proceedings of Lecture Notes in Computer Science, Computational Science–ICCS 2002, Part II, vol. 2330, 2002, pp. 824–833.

[21] K. Hinsen, The molecular modeling tool kit: a new approach to molecular simulations, J. Comput. Chem. 21 (2000) 79–85.

[22] J.E. Stajich, et al., The bioperl toolkit: Perl modules for the life sciences, Genome Res. 12 (2002) 1611–1618.

[23] B. Roux, T. Simonson, Implicit solvent models, Biophys. Chem. 78 (1999) 1–20.

[24] C.J. Cramer, D.G. Truhlar, Implicit solvation models: equilibria, structure, spectra, and dynamics, Chem. Rev. 99 (1999) 2161–2200.

[25] M.P. Allen, D.J. Tildesley, Computer Simulation of Liquids, first ed., Oxford University Press, New York, 1987.

[26] T.A. Darden, D. York, L.G. Pedersen, Particle mesh Ewald: an Nlog(N) method for Ewald sums in large systems, J. Chem. Phys. 98 (1993) 10089–10092.

[27] D. Bashford, D.A. Case, Generalized Born models of macromolecular solvation effects, Ann. Rev. Phys. Chem. 51 (2000) 129–152.

[28] A. Kolinski, J. Skolnick, Assembly of protein structure from sparse experimental data: an efficient Monte Carlo model, Proteins 32 (1998) 475–494.

[29] E.S. Huang, et al., Accuracy of side-chain prediction upon near-native protein backbones generated by ab initio folding methods, Proteins 33 (1998) 204–217.

[30] R.L. Dunbrack Jr., M. Karplus, Backbone-dependent Rotamer library for proteins: application to side-chain prediction, J. Mol. Biol. 230 (1993) 543–574.

[31] M.P. Jacobson, et al., Force field validation using protein side chain prediction, J. Phys. Chem. B 106 (2002) 11673–11680.

[32] Z.X. Xiang, B. Honig, Extending the accuracy limits of prediction for side-chain conformations, J. Mol. Biol. 311 (2001) 421–430.

[33] D. Comer, Internetworking with TCP/IP: Principles, Protocols, and Architecture, vol. 1, fourth ed., Prentice Hall, 2000.

[34] U.H.E. Hansmann, Parallel tempering algorithm for conformational studies of biological molecules, Chem. Phys. Lett. 281 (1997) 140–150.

[35] A.E. Garcia, K.Y. Sanbonmatsu, α-Helical stabilization by side chain shielding of backbone hydrogen bonds, Proc. Nat. Acad. Sci. U.S.A. 99 (2002) 2782–2787.

[36] Y. Sugita, A. Kitao, Y. Okamoto, Multidimensional replica-exchange method for free-energy calculations, J. Chem. Phys. 113 (2000) 6042–6051.

[37] N. Kresge, V.D. Vacquier, C.D. Stout, The crystal structure of a fusagenic sperm protein reveals extreme surface properties, Biochemistry 40 (2001) 5407–5413.

[38] M.J. Bower, F.E. Cohen, R.L. Dunbrack, Prediction of protein side-chain Rotamers from a backbone-dependent Rotamer library: a new homology modeling tool, J. Mol. Biol. 267 (1997) 1268–1282.

[39] M.S. Lee, F.R. Salsbury, Jr., C.L. Brooks, III., Novel generalized Born methods, J. Chem. Phys. 116 (2002) 10606–10614.

[40] M.S. Lee, et al., A new analytical approximation to the standard molecular volume definition and its application to generalized Born calculations, J. Comput. Chem. 24 (2003) 1348–1356.

[41] H.M. Berman, et al., The protein data bank, Nucl. Acids Res. 28 (2000) 235–242.

[42] B. Zagrovic, E.J. Sorin, V. Pande, β-hairpin folding simulations in atomistic detail using an implicit solvent model, J. Mol. Biol. 313 (2001) 151–169.

[43] X. Daura, et al., Reversible peptide folding in solution by molecular dynamics simulation, J. Mol. Biol. 280 (1998) 925–932.

[44] A. Hiltpold, et al., Free energy surface of the helical peptide Y(MEARA)$_6$, J. Phys. Chem. B 104 (2000) 10080–10086.

[45] Y. Pak, S. Wang, Folding of a 16-residue helical peptide using molecular dynamics simulation with Tsallis effective potential, J. Chem. Phys. 111 (1999) 4359–4361.

[46] S. Jang, S. Shin, Y. Pak, Molecular dynamics study of peptides in implicit water: ab inito folding of β-hairpin, β-sheet, and ββα-motif, J. Am. Chem. Soc. 124 (2002) 4976–4977.

[47] Y. Pak, S. Jang, S. Shin, Prediction of helical peptide folding in an implicit water by a new molecular dynamics scheme with generalized effective potential, J. Chem. Phys. 116 (2002) 6831–6835.

[48] W. Shalongo, L. Dugad, E. Stellwagen, Distribution of Helicity within the Model Peptide Acetyl(AAQAA)$_3$ amide, J. Am. Chem. Soc. 116 (1994) 8288–8293.

[49] A.G. Cochran, N.J. Skelton, M.A. Staovasnik, Tryptophan zippers: stable, monomeric β-hairpins, Proc. Nat. Acad. Sci. U.S.A. 98 (2001) 5578–5583.

[50] M. Nina, D. Beglov, B. Roux, Atomic radii for continuum electrostatics calculations based on molecular dynamics free energy simulations, J. Phys. Chem. 101 (1997) 5239–5248.

[51] M. Feig, A.D. MacKerell, Jr., C.L. Brooks, III, Force field influence on the observation of π-helical protein structures in molecular dynamics simulations, J. Phys. Chem. B 107 (2003) 2831–2836.

[52] A.T. Brünger, C.L. Brooks III, M. Karplus, Active-site dynamics of ribonuclease, Proc. Nat. Aacd. Sci. U.S.A. 82 (1985) 8458–8462.

[53] A. Fiser, et al., Evolution and physics in comparative protein structure modeling, Accounts Chem. Res. 35 (2002) 413–421.

[54] N. Rathore, J.J. de Pablo, Monte Carlo simulation of proteins through a random walk in energy space, J. Chem. Phys. 116 (2002) 7225–7230.

[55] W. Kabsch, C. Sander, Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features, Biopolymers 22 (1983) 2577–2637.

[56] A.D. MacKerell Jr., All-atom empirical potential for molecular modeling and dynamics studies of proteins, J. Phys. Chem. B 102 (1998) 3586–3616.