BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 21 2006, pages 2693–2694
doi:10.1093/bioinformatics/btl460

Structural bioinformatics

iFold: a platform for interactive folding simulations of proteins

Shantanu Sharma[1], Feng Ding[1], Huifen Nie[1], Daniel Watson[2], Aditya Unnithan[2], Jameson Lopp[2], Diane Pozefsky[2] and Nikolay V. Dokholyan[1,*]

[1]Department of Biochemistry and Biophysics and [2]Department of Computer Science,
University of North Carolina, Chapel Hill, NC 27599, USA

Received on August 14, 2006; revised and accepted August 23, 2006
Advance Access publication August 29, 2006
Associate Editor: Anna Tramontano

ABSTRACT

Summary: We built a novel web-based platform for performing discrete molecular dynamics simulations of proteins. *In silico* protein folding involves searching for minimal frustration in the vast conformational landscape. Conventional approaches for simulating protein folding insufficiently address the problem of simulations in relevant time and length scales necessary for a mechanistic understanding of underlying biomolecular phenomena. Discrete molecular dynamics (DMD) offers an opportunity to bridge the size and timescale gaps and uncover the structural and biological properties of experimentally undetectable protein dynamics. The iFold server supports large-scale simulations of protein folding, thermal denaturation, thermodynamic scan, simulated annealing and p_{fold} analysis using DMD and coarse-grained protein model with structure-based Gō-interactions between amino acids.

Availability: http://ifold.dokhlab.org

Contact: dokh@med.unc.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

One of the most challenging issues with studies of biological systems is the time and length scales that are relevant to biology. For example, some chemical reactions occur at the time scales of femtoseconds (10^{-15} s), while protein aggregation occurs at the time scales of hours and even years (10^4 s). Hence, the range of biologically relevant time scales spans 20 orders of magnitude. Similarly, the range of length scales that are of interest to biology spans over six orders of magnitude. None of the experimental, theoretical and computational approaches alone can probe these time and length scales as a whole. Dynamic and structural features of large biomolecules are often 'invisible' to current experimental techniques owing to their inherent resolution limitations in length and time scales. Computational approaches offer a unique opportunity to uncover the atomic structure and biological properties of experimentally challenging molecules and molecular complexes.

Direct computational approaches employing all-atom molecular dynamics (MD) simulations provide detailed information on the local dynamics of molecules. However, owing to the complexity of protein conformational space, all-atom MD simulations have severe limitations on the time and length scales that can be studied. An alternative approach is the simplification of protein models. In the simplified protein models, amino acids are coarse-grained to the level of effective beads (Dokholyan *et al*., 1998) The interaction potential between these beads can be derived from protein structure, *in vivo* experiments or biophysical analyses. A more realistic simulation approach for simplified protein models is discrete molecular dynamics (DMD). This approach permits the rapid and accurate sampling of the conformational space of biomolecules and their complexes (Ding and Dokholyan, 2005). One of the remarkable illustrations of the speed and accuracy of DMD approach is its ability to recapitulate the experimental studies of unfolded protein states and unravel their properties (Ding *et al*., 2005). Success of DMD approach in studies of proteins dynamics makes it a very valuable tool for the community of computational molecular biologists. The goal of this work is to bring DMD to the multidisciplinary community of bioinformatics researchers through the web (http://ifold.dokhlab.org). The goal of iFold is not protein structure prediction, but rather utilizing the native structure for deciphering protein dynamics. The simplicity of the iFold user-interface allows this server to be used also by experimentalists for probing possible molecular states.

2 METHODS

Hardware resources for large-scale DMD simulation jobs run by the iFold server are based on a 300-node Beowulf Linux cluster provided by the University of North Carolina. The underlying tool in iFold is of DMD (Dokholyan *et al*., 1998). The front end of iFold server consists of two subparts: (1) Client-side: The presentation layer that the user interacts with, using a web-browser; and (2) Server-side: The business logic part that processes the information the user inputs and aggregates information that needs to be sent back to the user (e.g. queue contents). The client-side is constructed in HTML and JavaScript and the server-side is built using PHP (http://www.php.net). The glue between the server-side and the client-side is the Smarty (http://smarty.php.net/) templating engine. Smarty allows creating HTML templates with cavities for PHP variables. At runtime, these cavities are filled in by PHP scripts, allowing an easy segregation between the presentation layer and the business logic. The server-side process interacts with the iFold scheduler–a Java application that verifies the user-specified inputs and submits simulations to iFold compute nodes using TCP-IP connections based on Java Sockets API.

Once a DMD simulation task is submitted to the iFold scheduler, it appends the simulation job to a pending jobs queue in which simulations are executed on a *first-come-first-serve* basis. As soon as an iFold compute node is available, the simulation's input parameters and desired outputs are dispatched to the compute node, over the Internet, using a Java socket connection. Upon successful completion of a DMD simulation, the compute

[*]To whom correspondence should be addressed.

node parses the list of desired outputs (simulation trajectory; p_{fold} value; graphs of energy versus temperature, energy versus time, or gyration radius versus time) and executes standard scripts for performing these analyses on DMD simulation results. The user is notified about the simulation summary via email and he/she may login to iFold web server to download the desired simulation results.

3 RESULTS

The key functions supported by the iFold front end are as follows:

- *Task Submission*: The iFold task submission process is driven by an XML file that holds all classes of simulation tasks available on the iFold server and the corresponding simulation parameters in a hierarchical manner. When the user loads the task submission page, the server-side PHP scripts parse the XML file, validating the parameters and filling the smarty templates for the client-side processing.
- *Registration Process*: To ensure security of the iFold server, human intervention is necessary for completion of the registration process. When the user registers on the main page, an email is automatically dispatched to his/her specified email address with a unique URL that contains a mathematically generated, encrypted key. When the user clicks the link, he/she is presented with a success page and another email is sent to an iFold administrator for approving the user's request for iFold.
- *Queue Management*: The activity page allows users to view their activity and the outputs generated by iFold. Users can see and delete only their own tasks; they can see if there are other tasks in the queue, but no information about the tasks. Administrators, on the other hand, can see all information on the queue and are able to delete tasks if necessary.

The iFold server has three modes of operation–

- The *guided user mode*: This mode is designed to provide a convenient user interface to biologists unfamiliar with simulation techniques. In this mode, simulations are performed using apposite default values for simulation parameters. Thus, by choosing a simulation task and specifying the structure of the protein, the users may run DMD simulations to collect relevant data such as melting temperature of their protein.
- The *advanced user mode*: This mode is designed for researchers familiar with various simulation techniques and gives the user freedom to specify valid ranges of simulation parameters and to download the simulation outputs in all formats supported by iFold.
- The *administrative user mode*: In this mode, the user is provided special privileges to have administrative access over other users of iFold. The regular operation of iFold server is fully automated; however, web-based support of multiple administrators for iFold is useful for providing expedited access for new users and monitoring new hardware requirements.

The iFold server is hosted at http://ifold.dokhlab.org and supports the following simulations. The input parameters for each of these simulations, their default values and the corresponding observables are described in the Supplementary Data.

(1) *Protein folding simulation*: The user enters the initial and final temperatures ($T_{\mathrm{init}}, T_{\mathrm{final}}$) at which the folding simulation of the protein is performed; starting from a linear conformation of the protein at temperature T_{init}, the temperature of the system is reduced at a constant, user-specified rate, until the system reaches the final temperature T_{final}.

(2) *Thermal denaturation*: Starting with the native protein structure at low temperature, the system's temperature is raised at a constant rate until the protein starts to unfold.

(3) *Thermodynamic scan*: Multiple constant temperature DMD simulations of the protein are performed over a range of temperatures to ascribe thermodynamic properties (such as heat, capacity and melting temperature) of the protein.

(4) *Simulated annealing*: The annealing process is iterated a number of times for effectively sampling the conformation space–rapidly raising temperature from the last stable conformation and relaxing protein at a slow rate. The lowest energy conformation approximates the folded protein.

(5) *Folding probability analysis*: p_{fold} measures the probability of a decoy to fold. It is a quantitative measure of progress in the folding pathway of the given conformation.

4 DISCUSSION

The architecture of iFold server is modular in nature and utilizes a Linux cluster as its compute resources. Adding more compute nodes to the system is feasible, thereby allowing the iFold server to scale up to an order of a million simulation tasks submitted. iFold's convenient user-interface is expected to make simulations accessible to molecular biologists for probing possible protein conformations, especially when conventional experimental techniques become unfeasible. Protein conformations featuring p_{fold} values close to 0.5 constitute the transition state ensemble. Combining results of DMD simulations from iFold with traditional molecular dynamics and quantum mechanics simulations will entail studies of proteins over vast time and length scales. Thus, the iFold server will enable effective sampling of biomolecular conformations, studying protein thermodynamics, kinetics and experimentally aided modeling.

ACKNOWLEDGEMENTS

We acknowledge the University of North Carolina Information Technology Services, especially Ruth Marinshaw, Steven Fishback and Paul Mitchell for hardware support. This work was supported by the Muscular Dystrophy Association grant MDA3720 and the American Heart Association grant no. 0665361U to N.V.D.

Conflict of Interest: none declared.

REFERENCES

Ding,F. and Dokholyan,N.V. (2005) Simple but predictive protein models. *Trends Biotechnol.*, **23**, 450–455.

Ding,F. *et al.* (2005) Scaling behavior and structure of denatured proteins. *Structure*, **13**, 1047–1054.

Dokholyan,N.V. *et al.* (1998) Discrete molecular dynamics studies of the folding of a protein-like model. *Fold Des.*, **3**, 577–587.