Journal of Computer-Aided Molecular Design **18:** 433–436, 2004.
© 2004 *Kluwer Academic Publishers. Printed in the Netherlands.*

Interactive essential dynamics

John Mongan
Bioinformatics Program; Medical Scientist Training Program; NSF Center for Theoretical Biological Physics, University of California at San Diego, La Jolla, CA 92093-0365, USA
(e-mail: jmongan@mccammon.ucsd.edu)

Received 13 May 2004; accepted in revised form 28 September 2004

Key words: essential dynamics, graphical user interface, interactive, molecular dynamics, principal component analysis, visualization

Summary

Essential dynamics (ED) is a useful method for analyzing trajectories generated by molecular dynamics (MD), but current tools are awkward to use, limiting the usefulness of the technique. This paper describes a new interactive graphical interface for visualization of ED results, including filtering a trajectory on an arbitrary set of eigenvectors and manipulation of a structure's projection along any eigenvector.

Abbreviations: ED – essential dynamics; IED – Interactive Essential Dynamics; MD – molecular dynamics; VMD – Visual Molecular Dynamics.

Introduction

Trajectories generated from molecular dynamics (MD) simulations provide a means to identify and study motions crucial for protein function [1]. Separating functionally important motions from random thermal fluctuations is a major challenge in analyzing MD trajectories. Principal component analysis of MD trajectory data, often called *essential dynamics* (ED) [2, 3], is frequently used to separate large-scale correlated motions from local harmonic fluctuations [4–9].

ED analysis constructs a new orthogonal basis set for the atomic coordinates in a trajectory, such that the greatest variance occurs along the first vector, with monotonically decreasing variance along successive vectors. These vectors are often called principal components or eigenvectors, since their derivation involves an eigen decomposition. The eigenvalues from the eigen decomposition represent the relative amount of molecular motion that occurs along each eigenvector. The eigenvalue spectrum is sharply peaked for molecular trajectory data, indicating that most of the molecular motion can be described by displacements along the first few eigenvectors [2, 4–6, 9]. A trajectory can be projected onto a subset of selected eigenvectors so only motion along the selected vectors is allowed. The most commonly selected subset is the first n eigenvectors such that a given percentage of the molecular motion occurs within the subspace formed by the selected eigenvectors. Projection onto these vectors filters out thermal noise, making the functionally interesting motions easier to appreciate. Smaller subsets may be selected to isolate a particular aspect of the molecule's motion. One can also examine the functional meaning of a single eigenvector by generating a trajectory with atomic positions interpolated between extreme projections on the selected eigenvector.

ED is a standard method of analysis that is widely implemented in molecular simulation packages [10–13]. Tools in these packages take trajectory and eigenvector files as input and produce a new trajectory as output, which must be loaded into an integrated [11] or external [10,

12, 13] viewer. A more flexible approach [14], implemented within a limited viewer, is not widely available. In the available tools, a separate trajectory file of interpolations between extreme projections must be generated for each eigenvector, and a separate filtered trajectory file must be generated for each set of eigenvectors selected for filtering. Some tools [11, 12] are limited to filtering along a single eigenvector at a time, which may be problematic since rotational motion cannot be adequately represented with a single eigenvector.

Generating and loading a separate trajectory file for each aspect of the ED results is cumbersome and discourages complete understanding of the ED analysis. Interactive essential dynamics (IED) is a new program that addresses these problems, providing fully interactive analysis of ED results through a graphical interface. Filtering eigenvectors can be rapidly added or removed from within the viewer, even while the trajectory is being played. The functional meaning of an eigenvector can be examined from within the viewer by dragging the atomic positions along the eigenvector using a slider control. Arrows representing an atom's motion along an eigenvector can be drawn to provide a static representation of an eigenvector, as in the work of Huitema and van Liere [14]. IED can calculate eigenvectors and projections directly, or read the results of calculations performed in GROMACS [10] or the ptraj module of AMBER 8 [13]. IED also allows sets of vectors that do not have accompanying projections to be loaded, so results of normal modes analysis performed by AMBER or GROMACS can be visualized. The Python scripting interface of visual molecular dynamics (VMD) [15] is used for display. The extensive visualization, animation, rendering and analysis capabilities of VMD remain available while using IED.

Theory and methods

To perform ED, coordinate data from each timestep is fitted to a reference structure to remove translational and rotational motion. The fitted trajectory data are used to construct a covariance matrix C according to Equation 1:

$$C = \left\langle (x - \langle x \rangle)(x - \langle x \rangle)^{\mathrm{T}} \right\rangle \tag{1}$$

where $\langle\,\rangle$ represents the mean across all timesteps, and the T superscript represents transpose. An eigen decomposition (or diagonalization) of the symmetric matrix C is performed to identify Λ, a diagonal matrix of eigenvalues and T, a matrix of column eigenvectors forming a new orthonormal basis set [2], satisfying

$$C = T \Lambda T^{\mathrm{T}} \tag{2}$$

A zero-mean trajectory matrix, X, can be constructed by subtracting $\langle x \rangle$ from the coordinate vector for each timestep to form the rows of X. The matrix of the projections of each timestep onto each eigenvector, P, is obtained by multiplying the trajectory matrix, X, by T

$$P = X T \tag{3}$$

For use with IED, these calculations may be performed using the AMBER or GROMACS suites. IED is also capable of performing these calculations itself, but is less efficient than AMBER or GROMACS.

The trajectory matrix, X, can be reconstructed from the eigenvectors and projection matrices T and P, by right multiplying Equation 3 by T^{T}:

$$P T^{\mathrm{T}} = X T T^{\mathrm{T}} = X I = X \tag{4}$$

where I is an identity matrix. More usefully, a matrix of filtered trajectory data, F, can be calculated by multiplying a subset of the (column) projection vectors in P by the corresponding subset of the eigenvectors in T^{T}. This way F contains only motions that occur along the eigenvectors selected from P and T, since motions along other eigenvectors are represented by projections omitted from the calculation of F. IED employs this method to calculate filtered trajectories, adding $\langle x \rangle$ to the coordinate vector in each row of F to translate the coordinates back to their original origins. When a single eigenvector is to be examined by interpolation between extreme projections, coordinates are calculated by varying the (scalar) projection value for the selected eigenvector at the current time step and recalculating the appropriate row from F for each value of the projection.

When IED calculates ED directly, it can operate on trajectory data in any format that VMD is able to load. When loading results of ED analysis carried out in GROMACS, it requires a molecular topology file (in any VMD acceptable format), an

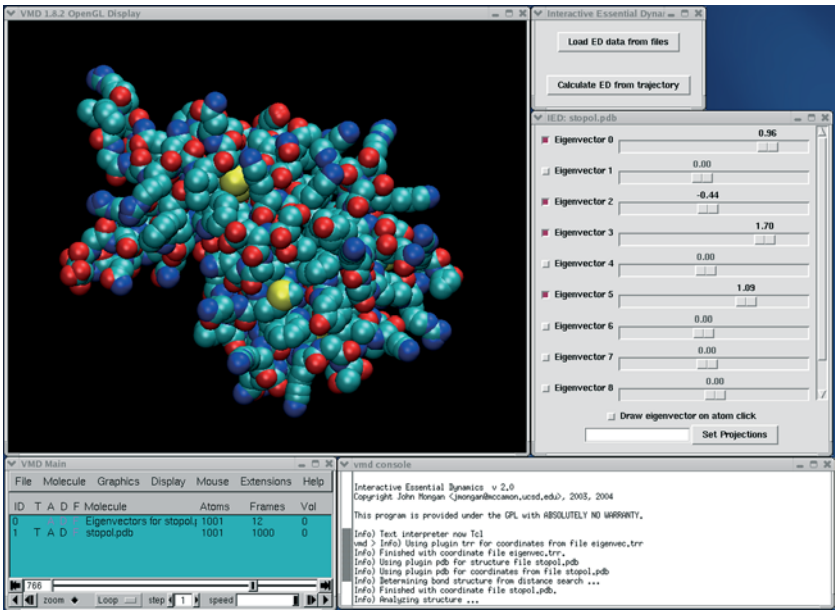

Figure 1. Screen shot of IED. Top right window is the main IED window, window immediately below contains the checkboxes and sliders for selecting eigenvectors and manipulating projections. Remaining windows are VMD windows: main window and animation controls at bottom left, console at bottom right and molecular display at top left.

eigenvectors file in GROMACS TRR binary format generated by g_covar, and a projections file generated by g_anaeig. The first timestep of the eigenvectors file is ignored, the second contains the molecule's average coordinates over the trajectory and the remaining timesteps contain eigenvectors in decreasing order of their eigenvalues. The projections file is formatted as text input to Grace, a plotting tool. Each eigenvector has a separate block of projection data within the file; within each block there is one projection per line, consisting of a time value followed by a projection value, separated by whitespace. When loading ED results from AMBER, the requirements are similar: a topology file, an eigenvectors file and a projections file. The eigenvectors file and projections file are both produced by ptraj and are in text format. The eigenvectors file contains two header lines, which are ignored, and the average coordinates, followed by the eigenvectors. Each eigenvector has a two line header consisting of a line containing 4 asterisks (****), followed by a line giving the ordinal number of the eigenvector and its eigenvalue. Numeric data for the average coordinates and eigenvectors are whitespace delimited, with 7 values per line. The projections file has a two line header which is ignored. Each successive line contains a timestep number, followed by projection values onto each eigenvector for that timestep. The values are whitespace delimited.

Internally, IED represents the eigenvector data in a VMD trajectory object and the projection data in a Python Numeric array object. IED is an open source application, and is easily extended to other file formats by writing parsing routines to read data into the aforementioned data structures.

User interface

IED is started either by selecting a trajectory in VMD for ED analysis, or by loading files containing the results of an ED analysis previously performed in ptraj or GROMACS. Once the ED data are loaded, a window is displayed with a checkbox and slider for each eigenvector (see Figure 1). When necessary, the eigenvector slider area of the window can be scrolled to allow for arbitrarily large numbers of eigenvectors. Selecting a checkbox allows motion along the corresponding eigenvector and activates the eigenvector's slider, setting its position to the projection on the eigenvector for the current frame of the trajectory. Check boxes can be selected independently,

allowing simultaneous analysis of any combination of eigenvectors. When the VMD animation controls are used to play the trajectory, the molecular display shows the filtered trajectory: the projection of the trajectory on the currently selected eigenvectors. Slider positions corresponding to selected eigenvectors are updated as each frame of the trajectory is displayed. The movement of the sliders provides an animated, graphical representation of the projection of the trajectory on each eigenvector. When the animation is stopped, the sliders for any selected eigenvector can be moved manually, which temporarily changes the projection value on the eigenvector for the displayed frame. The molecular display is updated as the slider is moved, making it easy to appreciate any eigenvector's contribution to the molecular motion. A comma delimited list of projections can be entered in the text box near the bottom of the window to rapidly set the projections along all eigenvectors.

Interactive manipulation of the molecule's projection along an eigenvector provides the clearest visualization of the eigenvector, but is not possible in cases where a static image is required for publication or presentation. Static visualizations can be produced by selecting a single eigenvector and clicking on representative atoms. An arrow is drawn through the clicked atoms, with the arrow's head representing the atom's position at the most positive projection and the tail representing the most negative projection.

When IED is used to visualize normal modes data, there is no associated trajectory, and no projections file is loaded. In this case trajectory playing and filtering features are disabled, but all other features are available.

Conclusions

IED allows interactive visualization and manipulation of projections of protein motion on selected eigenvectors and easy selection and filtering on different discontinuous sets of eigenvectors. It increases efficiency in working with ED results and enables appreciation of aspects of the dynamics that might be missed with more limited tools.

IED is freely available under the Gnu Public License at http://mccammon.ucsd.edu/software.html. The language, applications and libraries on which it depends are also freely available.

Acknowledgements

I thank Prof. J. Andrew McCammon for helpful suggestions and Dr. Justin Gullingsrud for advice on VMD and Python. I thank the Taft family for their support through the Taft Fellowship, and The Burroughs Wellcome Fund for their support through the La Jolla Interfaces in Science Predoctoral Fellowship. This work has been supported in part by grants from NSF, NIH, the NSF Center for Theoretical Biological Physics, the National Biomedical Computing Resource, and Accelrys, Inc.

References

1. Shen, T., Tai, K. Henchman, R.H. and McCammon, J.A., Acc. Chem. Res., 35 (2002) 332.
2. Amadei, A., Linssen, A.B.M. and Berendsen, H.J.C., Proteins: Struct. Funct. Genet., 17 (1993) 412.
3. García, A.E., Phys. Rev. Lett., 68 (1992) 2696.
4. Hamelberg, D., Mongan, J. and McCammon, J.A., J. Chem. Phys., 120 (2004) 11919.
5. de Groot, B.L., Daura, X., Mark, A.E. and Grubmuller, H., J. Mol. Biol., 309 (2001) 299.
6. Xiong, B., Huang, X.Q., Shen, L.L., Shen, J.H., Luo, X.M., Shen, X., Jiang, H.L. and Chen, K.X., Acta Pharmacol. Sin., 25 (2004) 705.
7. Lee, J., Suh, S.W. and Shin, S., J. Biomol. Struct. Dyn., 18 (2000) 297.
8. Crabbe, M.J., Cooper, L.R. and Corne, D.W., Comput. Biol. Chem., 27 (2003) 507.
9. Arcangeli, C., Bizzarri, A.R. and Cannistraro, S., Biophys. Chem., 90 (2001) 45.
10. Lindahl, E., Hess, B. and van der Spoel, D., J. Mol. Model., 7 (2001) 306.
11. Vriend, G., J. Mol. Graph., 8 (1990) 52.
12. Kendall, R.A., Apra, E., Bernholdt, D.E., Bylaska, E.J., Dupius, M., Fann, G.I., Harrison, R.J., Ju, J., Nichols, J.A., Nieplocha, J., Straatsma, T.P., Windus, T.L. and Wong, A.T., Comput. Phys. Commun., 128 (2000) 260.
13. Case, D., Darden, T., Cheatham III, T.E., Simmerling, C., Wang, J., Duke, R., Luo, R., Merz, K., Wang, B., Pearlman, D., Crowley, M., Brozell, S., Tsui, V., Gohlke, H., Mongan, J., Hornak, V., Cui, G., Beroza, P., Schafmeister, C., Caldwell, J., Ross, W. and Kollman, P., AMBER 8, 2004.
14. Huitema, H. and van Liere, R., IEEE Visualization 2000 Proceedings 11, 2000.
15. Humphrey, W., Dalke, A. and Schulten, K., J. Mol. Graph., 14 (1996) 33.